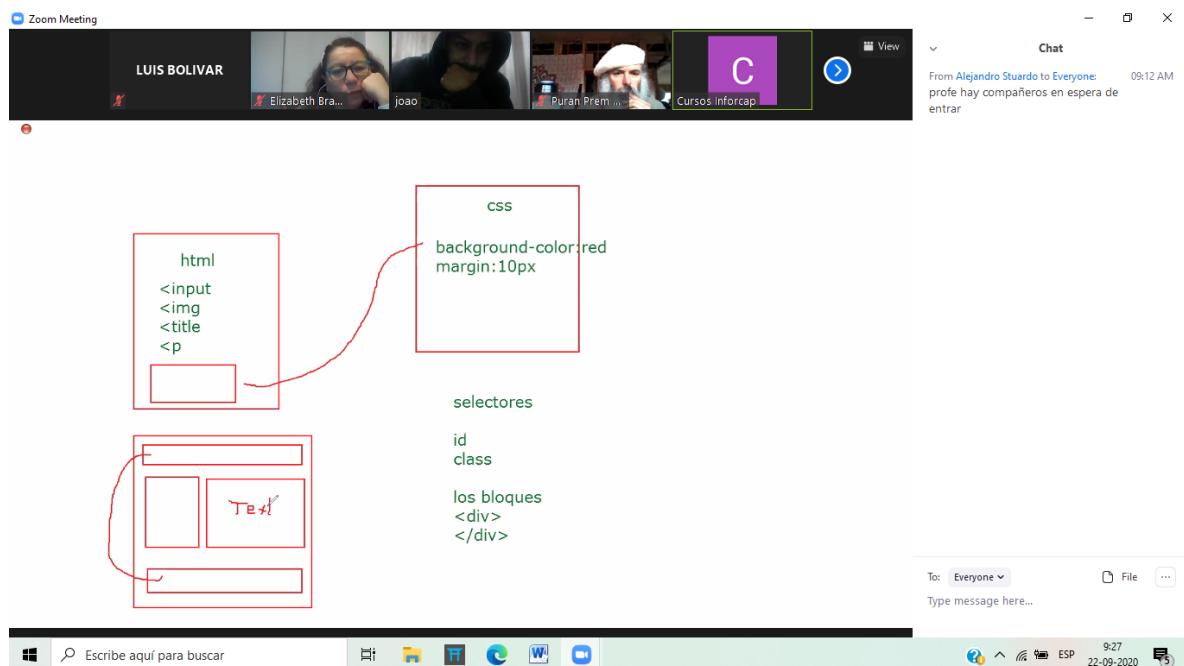


## Css hoja de estilo en cascada

Selectores (id , class)y bloques( <div><'>\div>) ordenan

Los div se van configurando en css

Css tiene extensión .css



## Inserción de css en head

The image is a screenshot of a web browser displaying a learning platform. The URL is 'learn.codingdojo.com'. The sidebar on the left is titled 'Empezando' and lists topics such as 'CSS Selector', 'Elementos de Estilo', 'Texto de Estilo', 'Inspeccionar Elemento', 'La Propiedad Display- Block...', 'Modelo de Caja (Box Model...)', 'Prueba', 'Concepto Lego', 'Trazando tus Bloques', 'La Propiedad de Posición (P...', 'Compatibilidad del Navegad...', and 'Preguntas Frecuentes'. The main content area shows an HTML file named 'css\_overview.html' with the following code:

```
1  <!DOCTYPE HTML>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>CSS Overview</title>
6      <meta name="description" content="A really cool site!">
7      <link rel="stylesheet" type="text/css" href="css_1.css">
8  </head>
9  <body>
10 </body>
11 </html>
```

A callout box in the bottom right corner says 'Pero así es como se hace.' The browser interface includes a search bar, a toolbar with icons for file operations, and a status bar at the bottom showing '10:00 22-09-2020'.

Línea 7 está la inserción

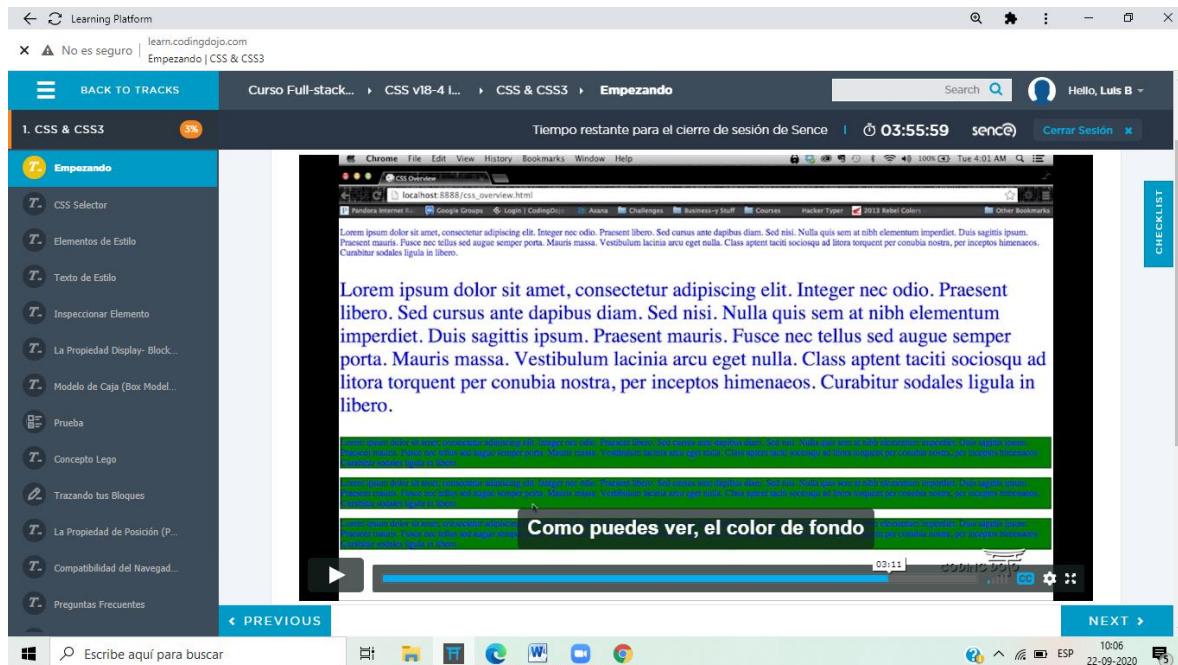
Este va al archivo css

The screenshot shows a web-based learning environment for CSS. On the left, a sidebar lists various CSS topics: CSS Selector, Elementos de Estilo, Texto de Estilo, Inspeccionar Elemento, La Propiedad Display- Block..., Modelo de Caja (Box Model...), Prueba, Concepto Lego, Trazando tus Bloques, La Propiedad de Posición (P...), Compatibilidad del Navegad..., Preguntas Frecuentes. The main area has tabs for 'css\_overview.html' and 'css\_1.css'. The code in 'css\_1.css' is:

```
1 p{ color:blue;
2 }
3 #important{
4     font-size:36px;
5 }
6 .info{
7     background-color: green;
8     border:1px solid black;
9 }
10 }
```

A central banner says 'y cómo dar formato en tu archivo CSS'. Below the code editor is a toolbar with icons for file operations. At the bottom, there are 'PREVIOUS' and 'NEXT' buttons, along with a search bar and system status indicators.

Y el ej se ve



Hay 3 formas de adjuntar CSS a un documento: *inline*, *internal*, y *external*. **Inline e Internal son consideradas malas prácticas.** Solo utilizaremos *external* stylesheets, enlazadas desde el documento HTML utilizando la etiqueta <head> tag. Revisa el curso de HTML si no recuerdas cómo hacer esto.

CSS utilizada un esquema de prioridad para determinar qué reglas de estilo aplicar si más de una regla encaja con un elemento particular. En este esquema, las prioridades o pesos ("weights") bajan en cascada y son asignadas a reglas, por lo que los resultados son predecibles.

Una stylesheet es una lista de **reglas** (rules). Cada regla (o "rule-set") consiste en uno o más **selectores** (selectors) y un **bloque de declaración** (declaration block).

## CSS Selector

Los *selectores* se utilizan para declarar los elementos de HTML para los cuales aplicará un estilo.

Los selectores más comunes son:

- Todos los elementos con una etiqueta específica de HTML (ej: p, h1)
- Los elementos especificados de acuerdo a los siguientes atributos:
  - **id** (son precedidos por # en CSS)
    - Debe ser único y solo puede ser utilizado una vez en tu página.
  - **class** (son precedidos por . en CSS)
    - Pueden ser utilizados muchas veces para compartir código CSS repetido.

### Bloque de Declaración

Un bloque de declaración es una lista de declaraciones entre llaves "{}". Cada declaración individual consiste en una **propiedad**, dos puntos (":"), y un **valor**. Si un bloque tiene múltiples declaraciones, tienen que estar separadas por un punto y coma (",").

Algunos ejemplos:

Para todas las etiquetas "p" de HTML, haz el color de fuente azul (blue).

```
p {  
    color: blue;  
}
```

Para un elemento con el id “important”, haz el color de fuente (font size) de 36px.

```
#important {  
    font-size: 36px;  
}
```

Para todos los elementos con la clase “info”, haz el fondo (background) verde y agrega un borde negro (black border) de 1px de ancho.

```
.info {  
    background-color: green;  
    border: 1px solid black;  
}
```

En estos ejemplos, **p**, **#important** e **.info** son **selectores**. **Color**, **font-size**, **background-color** y **border** son **propiedades**.

### Más sobre selectores:

- <http://www.webdesignerdepot.com/2013/08/10-css-selectors-you-shouldnt-code-without/>
- [http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)

Ahora que sabemos un poco más sobre cómo usar selectores, hagamos una lista general de las propiedades internas para CSS:

1. element (1 punto)
2. .class (10 puntos)
3. #id (100 puntos)

Este es el orden de prioridad predeterminado. Adicionalmente, la especificidad también cuenta, por ejemplo, **ul li** anulará a **li**. W3C ha creado una buena tabla respecto a cómo deberías calcular el peso interno (internal weight):

```
li          {...} /* a=0 b=0 c=1 -> specificity = 1 */  
ul li       {...} /* a=0 b=0 c=2 -> specificity = 2 */  
ul ol li    {...} /* a=0 b=0 c=3 -> specificity = 3 */  
li.red      {...} /* a=0 b=1 c=1 -> specificity = 11 */  
ul ol li.red {...} /* a=0 b=1 c=3 -> specificity = 13 */  
#list       {...} /* a=1 b=0 c=0 -> specificity = 100 */
```

- **a** representa el número de atributos #id en el selector
- **b** representa el número de atributos de clase (class)

- `c` representa el número de nombres de etiqueta

Combinar esto en un número te dará el peso real. Esto significa que `li #list` tendrá el mismo peso que `ul #list`. Esto es, probablemente, una de las cosas más confusas del esquema en cascada (cascade scheme), pero es de hecho más simple de lo que piensas: **se trata de contar.**

## Elementos de Estilo

Más abajo se presentan algunos de las propiedades de estilo más comunes que afectarán a todos los elementos. No es una lista definitiva y debieras buscar otras propiedades que puedan ayudarte a agregar estilo a tus documentos.

### **width | height:**

Las propiedades width (ancho) y height (altura) son utilizadas para determinar el tamaño de tus elementos. Los valores pueden ser expresados en pixeles (px) y porcentaje (%). Cuando se trabaja con contenido estático, es recomendable utilizar px puesto que definirás tu página para que no cambie. Si trabajas con un diseño responsivo, es mejor utilizar %,

Ten cuidado al establecer tus propiedades de altura ya que esto determinará cuánto contenido puede tener el elemento. Si no lo configuras, tu elemento se extenderá para encajar con el contenido. Por otro lado, si lo configuras y tienes más contenido que el que se puede mostrar, entonces tendrás que ajustar la altura manualmente cada vez que cambies el contenido o deberás utilizar la propiedad overflow.

Cuando quieras redimensionar tu elemento (ej: imágenes), puedes ajustar una propiedad (ancho o altura) y la otra se ajustará automáticamente para conservar las dimensiones relativas de la imagen.

```
a {  
    width: 25px;  
}  
div {  
    width: 100%;  
    height: 200px;  
}  
img {  
    width: 250px;  
}
```

## **overflow:**

Esta propiedad determina qué debiera pasar cuando el contenido dentro de un contenedor es demasiado para el tamaño del contenedor. Puedes configurar el contenedor para ocultar la información adicional que no cabe, mostrar la información como sea o agregar una barra de desplazamiento (scroll bar) con lo que el contenido podría ser visto pese a su tamaño.

```
div {  
    overflow: scroll;  
}
```

## **background:**

La propiedad background (fondo) puede modificar el fondo de un elemento en una sola línea, lo que es más sencillo que separar cada propiedad en su propia línea de código. El color puede ser definido utilizando hex, rgb o código semántico (semantic code).

```
p {  
    background: #ffffff url("cherries.png") no-repeat fixed center;  
}
```

**background-color | background-image | background-position | background-size |**

## **background-repeat:**

Estas propiedades ajustan el fondo de acuerdo a su tipo. Como background, background-color puede ser definida utilizando hex, rgb o código semántico.

```
p {  
    background-color: blue;  
}  
div {  
    background-image: url("cherries.png");  
    background-position: center;  
    background-size: auto;  
    background-repeat: no-repeat;  
}
```

## **border:**

Esta propiedad ajusta todos los elementos de borde en una sola línea. Así, el primer valor es el grosor del borde. El segundo valor es tipo de borde, mientras que el tercero es el color del borde. El color puede ser hex, rgb o código semántico.

La propiedad border puede separarse en distintas líneas utilizando border-width, border-style y border-color. Además, puedes seleccionar específicamente qué borde modificar utilizando border-top (arriba), border-bottom (abajo), border-right (a la derecha) y border-left (a la izquierda).

```
button {  
    border: 2px dotted green;  
}  
div {  
    border: 1px solid #000000;  
}  
p {  
    border-right: 1px groove rgb(100,100,100);  
    border-left: 1px groove rgb(200,200,200);  
}
```

### **border-radius**

Esta propiedad permite dar una apariencia curva a las esquinas de tu borde. Los valores puede ser establecidos utilizando px o %.

```
button {  
    border-radius: 5px;  
}
```

# **Texto de Estilo**

A continuación algunas de las más comúnmente utilizadas propiedades de estilo de texto (text style properties). Esta no es la lista completa, por lo que te invitamos a buscar en internet otras propiedades de estilo de texto..

### **color:**

El color de texto se especifica usando la propiedad de color. El valor de la propiedad de color puede ser expresada utilizando hex, RGB o código semántico.

```
p {  
    color: #ffffff;  
}  
a {  
    color: rgb(255, 255, 255);  
}  
span {  
    color: white;  
}
```

### **text-align:**

Esta propiedad se utiliza para establecer la alineación horizontal de cualquier texto. El texto puede estar centrado, alineado a la izquierda o a la derecha, o justificado. Esta propiedad solo funcionará si también presenta un bloque.

```
h1 {  
    text-align: center;  
}
```

### **text-decoration:**

Es utilizada para agregar o remover underlines (subrayados), overlines (línea sobre) y line-throughs (tachado).

```
a {  
    text-decoration: none;  
}
```

## **Font (Fuente)**

### **font-family:**

La propiedad font-family especifica el estilo de fuente a ser usado por un elemento. Existen 2 tipos de nombres de font-family:

- named-family - Ejemplos: "times", "courier", "arial"
- generic-family - Ejemplos: "serif", "sans-serif", "cursive", "fantasy", "monospace"

### **Fuentes Web-safe:**

Existen fuentes instaladas en casi todos los sistemas y se denominan web safe fonts (fuentes seguras para la web), porque en general se verán igual independiente del sistema que tenga el usuario. Algunas de ellas son: Verdana, Arial, Trebuchet MS, Times New Roman, Georgia, Andale Mono, Courier New, Comic Sans e Impact.

```
p {  
    font-family: "helvetica neue", arial, verdana, sans-serif;  
}
```

Font-family permite ingresar múltiples opciones cuando el navegador las recorre de izquierda a derecha hasta que encuentra una fuente en el sistema que puede ser usada. Esto significa que siempre debes asegurarte de tener una opción generic-family como tu último valor, ya que si alguno de los otros puede ser encontrado, el navegador utilizará la opción predeterminada para la familia especificada.

- El navegador busca la fuente Helvetica Neue en el sistema de usuario y si la encuentra la utiliza.

- Si no encuentra Helvetica Neue, el navegador busca Arial en el sistema del usuario y si la encuentra la utiliza.
- Si Arial no es encontrada, usará Verdana.
- Como último recurso, si no encuentra ninguna de las fuentes en el grupo de fuentes, volvemos a sans-serif instruyendo al navegador que use sea cual sea la fuente sans-serif predeterminada del sistema. No sabrás que se usará exactamente en este caso, pero al menos es mejor que terminar con la fuente predeterminada del navegador, Times new roman, la cual es una fuente serif.

**Ten en cuenta que las fuentes con más de una palabra en su nombre deben ir entre comillas.**

### **font-size:**

La propiedad font-size (tamaño de fuente) puede ser expresada en 4 unidades diferentes: pt, px, em, %.

Pt y px (punto y pixel) son consideradas dimensionamiento estático y no se ajustarán adecuadamente cuando se redimensione tu página. Em y % (medida responsive y porcentaje) sí se ajustarán y, por ende, son las recomendadas por los desarrolladores. A continuación podrás ver un cuadro desplegable que muestra las equivalencias aproximadas entre las cuatro unidades.



```

h3 {
  font-size: 10pt;
}
p {
  font-size: 14px;
}
a {
  font-size: 1.2em;
}
span {
  font-size: 80%;
}
  
```

### **font-style**

Esta propiedad controla la inclinación de la letra. Algunos textos pueden tener la propiedad italic (cursiva) como parte de la fuente, por lo que seleccionarla sería simple. Para las fuentes que no tienen el estilo italic, el valor oblique puede ser utilizado para imitar el texto en italic.

```

span {
  font-style: italic;
}
  
```

## **font-weight**

Esta propiedad define el grosor de una línea de caracteres. normal es normalmente el valor por defecto.

Los valores pueden ser establecidos de forma numérica o semántica.



```
p {  
    font-weight: bold;  
}  
span {  
    font-weight: 600;  
}
```

## **¿Qué es una etiqueta span?**

Si quieras resaltar un texto específico dentro de un párrafo o división, entonces puedes <span>. A diferencia de <div> que sitúa al siguiente <div> en una línea separada, <span> por defecto pone al elemento en la misma línea. Por ejemplo, considera

```
<h1>Hello <div>World</div> </h1>
```

Arriba por defecto pondríamos ‘World’ (Mundo) en una nueva línea. Sin embargo, si quisieras que World se mostrara en la misma línea, deberías escribir

```
<h1>Hello <span>World</span> </h1>
```

Luego, en el css, podrías hacer algo como esto para resaltar solo la palabra "World":

```
h1 span {  
    font-weight: bold;  
    color: blue;  
}
```

Para leer más sobre la diferente entre <div> y <span>, puede

leer: [https://www.codecademy.com/en/forum\\_questions/502ad0ea558dfe0002026d69](https://www.codecademy.com/en/forum_questions/502ad0ea558dfe0002026d69)

# **Cómo usar Inspeccionar Elemento (Inspect Element)**

Editar HTML y CSS puede verse como desactivar una bomba, un pequeño cambio y tu proyecto enloquece. Tu mejor aliado al editar son las herramientas de desarrollador (Developer Tools). Se incluyen en **Chrome, Firefox, la última versión de Internet Explorer (Edge) y Safari**.

Para acceder a Inspeccionar Elemento, simplemente usa el clic derecho en cualquier elemento de la ventana del navegador y cliquea en Inspeccionar (**Inspect**) ¡y listo!. Inspeccionar Elemento es una característica que muestra todas las propiedades del sitio web que estás viendo.

The screenshot shows the Chrome DevTools interface with the 'Elements' tab selected. The main area displays the HTML structure of a page from ESPN. The code includes various script tags, div elements with IDs like 'scoreboard' and 'content', and comments indicating browser-specific code for IE 8. At the bottom of the code block, there is a note: '<!-- SiteCatalyst code version: H.21.3 Copyright 1997-2010 -->'. Below the code, the toolbar shows icons for copy, paste, find, and search, followed by the file name 'html.js' and the URL 'body.frontpage.espn360.espn3'. A blue box highlights the 'div.bg-elements' entry in the dropdown menu.

```
> <script type="text/javascript">...</script>
> <div id="scoreboard">...</div>
<!-- end scoreboard -->
<script>espn.core.init.scoreboard();</script>
<!-- end scoreboard -->
<!-- begin content -->
> <div id="content" class="container">...</div>
<!-- end content -->
<!-- begin footer -->
> <div id="footer" class="container">...</div>
<!-- end footer -->
<!--[if lte IE 8]></div><![endif]-->
</div>
> <script type="text/javascript">...</script>
<!-- begin inv -->
<!-- end inv -->
<!-- begin popunder -->
<!-- end popunder -->
<!-- SiteCatalyst code version: H.21.3 Copyright 1997-2010 -->
```

Esta es una captura de pantalla de la ventana inspeccionar de Chrome. En la barra gris de arriba, verás un conjunto de opciones incluyendo ‘elements’ y ‘console’, que muestran distinta información de la página.

## Pestaña de Elementos (Elements Tab)

Por las siguientes dos lecciones, nos centraremos en ESPN.com. Puesto que la mayoría de las empresas cambian sus sitios webs con frecuencia, ten en cuenta que los sitios de ESPN que verás, pueden ser un tanto diferentes a los ejemplos del vídeo de más abajo. Pese a ella, las mismas herramientas y conceptos pueden ser aplicados.

La pestaña **elements**, te permite ver el HTML que está siendo procesado por tu navegador. Puedes ver todas las *classes* e *ids* del contenido HTML, lo que es de gran ayuda para depurar CSS. La imagen de más arriba, muestra la pestaña de elementos de una porción del HTML de ESPN.com ¡Fíjate que incluso puedes ver comentarios!!

Si cliqueas en un elemento particular del HTML dentro del cuadro de **elementos**, verás que el lado derecho de la ventana de Inspeccionar Elemento cambia. Esta parte de la ventana muestra una tonelada de información de **CSS y JavaScript**. Hay una pestaña a la derecha llamada ‘estilos’ (styles) que es por lejos la herramienta más útil al trabajar con CSS.

The screenshot shows two instances of the Chrome DevTools Styles tab. The top instance is for the element `#header .espn-logo span, #header .espn-logo-lrg a`. It displays the following CSS rules:

```

element.style {
}
media="screen"          (index)
.frontpage   c?v=20131107100...2.css,modul...:2
#header .espn-logo span, #header .espn-
logo-lrg a {
    width: 110px;
    height: 47px;
    margin-right: 10px;
    float: left;
    text-indent: -1000px;
}
media="screen"          (index)
html, body,  c?v=20131107100...2.css,modul...:2
div, span, applet, object, iframe, h1, h2,
h3, h4, h5, h6, p, blockquote, pre, a,
abbr, acronym, address, big, cite, code,

```

The bottom instance is for the element `caption, tbody, tfoot, thead, tr, th, td`. It displays the following CSS rule:

```

caption, tbody, tfoot, thead, tr, th, td {
    font-size: 100%;
}

```

Both instances show a detailed box model diagram for the ESPN logo element. The diagram illustrates the padding (green), border (yellow), and margin (orange) around the content area (blue). The content area has dimensions of 110 x 47 pixels, and the margin is 10 pixels on all sides.



(Este es el elemento que detallan las 2 imágenes anteriores)

La pestaña **estilos** muestra todo el CSS procesado para un elemento particular. La captura de pantalla anterior es la información de CSS para el logo de ESPN en su página web. A la izquierda de la imagen se muestra los estilos ejecutados del elemento y la stylesheet de donde fueron tomados. Baja hasta el final de la pestaña y verás la imagen a la derecha, que debería ser familiar. Esta imagen representa a los valores del modelo de cajas del elemento en cuestión. Puedes ver el **margen (margin)**, **relleno (padding)**, **borde (border)** y las dimensiones de cualquier elemento de la página. Aún mejor, puedes pasar por una parte específica de la caja en la ventana de inspeccionar elemento y la propiedad particular del elemento será resaltada en la página ¡Inténtalo! Esta herramienta es genial para ajustar la posición (los cambios solo se realizarán en el navegador y no se guardarán en el archivo CSS)

Mira la imagen nuevamente, verás un sector donde dice ‘**element.style**’. Ahí puedes escribir tu propio CSS y modificar lo que se muestra en la página web. Por supuesto, estos cambios no son permanentes, pero ya puedes imaginarte lo útil de esta herramienta. Otra cosa más que puedes hacer con todas las reglas de estilos de la pestaña estilos (styles), es desactivar las propiedades CSS asociadas desmarcándolas.

## Cómo usar Inspeccionar Elemento Genial, ¿y ahora qué?

Cualquier buen desarrollador web te dirá que **Inspeccionar Elemento** es una excelente herramienta en la lucha contra el código deficiente. Hay personas cuyo trabajo es específicamente hacer estas herramientas de desarrollo, por lo que estás en buenas manos. Inspeccionar es muy fácil de entender cuando la exploras ¡Empieza a usarla regularmente y te ahorrará mucho tiempo!

## Selector de Color

El diseño y selección de la paleta de colores representan una parte importantes en la creación de una página web. Para ayudar con una selección de colores precisa, hay una variedad de herramientas disponibles para encontrarlos y definirlos colores, además de armar paletas de colores. Es fácil encontrar en internet herramientas para generar esquemas de colores, pero encontrar el color exacto utilizado en el borde del botón de tu sitio web favorito puede ser más complicado. En esta sección esbozaremos cómo usar la herramienta de selector de color (colorpicker) de Chrome Dev, que nos ayudará a encontrar colores que nos interesan desde otros sitios web.

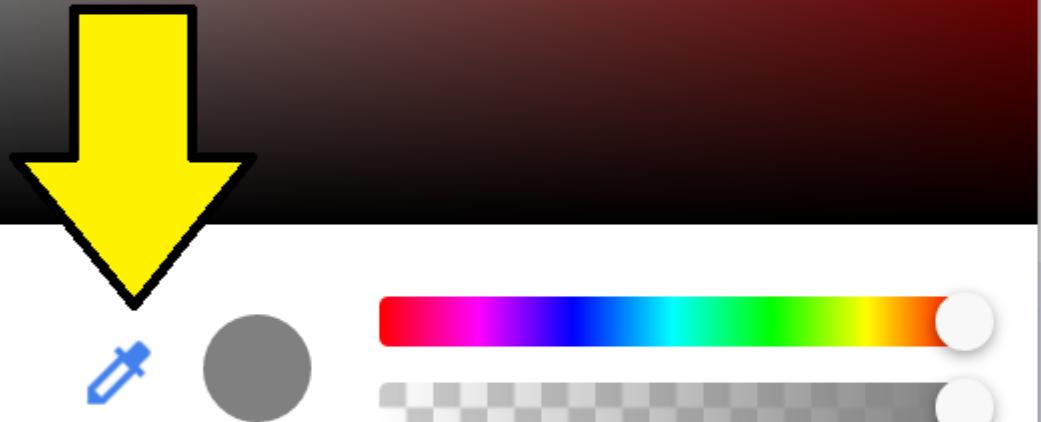
Para acceder al selector de color, primero necesitamos abrir las herramientas de Chrome Dev y seleccionar un elemento con un color de fondo. Luego, cliqueamos en el cuadrado de color de la ventana de Estilo (Styles). Entonces, se mostrará información respecto al color de fondo del elemento, permitiendo distintos ajustes como los valores del código de colores, distintas opciones de colores y una gradiante de intensidad y oscuridad. Desde ahí, es posible encontrar el valor de los colores en la página moviendo el cursor del mouse por la página. Por medio de un círculo se mostrarán los píxeles ampliados. El píxel central del círculo es el que toma el valor del color al hacer clic.



Elements Console Sources Network

```
><span class="control control-button">
><div class="control control-button">
><span class="control control-button">
><div class="control control-wheel" style="height: 400px;">
    <div class="wheel-bgcol" style="width: 107px; height: 107px;"></div>
    <div class="wheel-tile tile-1" style="background-color: #808080; width: 107px; height: 107px;"></div>
    <div class="wheel-tile tile-2" style="background-color: #808080; width: 107px; height: 107px;"></div>
    <div class="wheel-tile tile-3" style="background-color: #808080; width: 107px; height: 107px;"></div>
    <div class="wheel-tile tile-4" style="background-color: #808080; width: 107px; height: 107px;"></div>
<div class="clickable over" style="background-color: #808080; width: 107px; height: 107px;"></div>
```

Toggles the color picker



html body

Styles

#808080

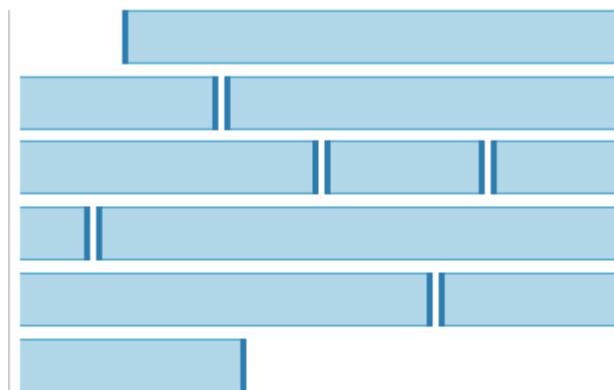
HEX

# La Propiedad Display- Block, Inline-Block, Inline

\*Puedes bajar una copia del código demostrado en el vídeo anterior desde el siguiente enlace [aquí \(block-inline.zip\)](#).

Cada elemento HTML es por defecto inline o block.

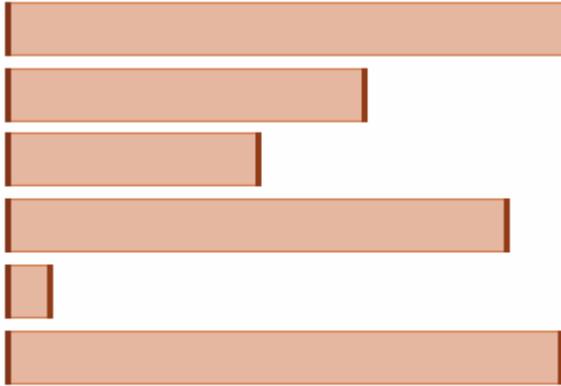
Los ítems *Inline* solo toman cuanto espacio sea necesario.



Inline elements flow from one line to the next.

Elementos Inline: <a>, <img>, <span>, <input>, <label>, <select>, <textarea>.

Los ítems **Block** toman el 100% del ancho del elemento parent (parent). Incluso si el ancho del elemento fuera menor que el parent, de todas formas abarcaría el 100% del ancho.



Block elements stack,  
regardless of their widths.

Elementos Block: <body>, <div>, <form>, <h1>-<h6>, <p>, <table>, <ul>, <ol>, <li>

**No es válido poner elementos block dentro de elementos inline.**

Ej:

No válido:

```
<a href="http://www.google.com">http://www.google.com</a>">www.google.com<h1>  
Google</h1></a>
```

Válido:

```
<h1><a href="http://www.google.com">http://www.google.com</a>">Googl  
e</a></h1>
```

Sin embargo, podemos cambiar el estilo de visualización predeterminado de HTML utilizando CSS y la propiedad de visualización (display). Los posibles valores de esta propiedad son: **inline**, **block**, **none** e **inline-block**.

Ej: Enlaces Inline predeterminados

[One](#)    [Two](#)    [Three](#)    [Four](#)    [Five](#)

```
<a href="#">One</a>  
<a href="#">Two</a>  
<a href="#">Three</a>  
<a href="#">Four</a>  
<a href="#">Five</a>
```

```
a {  
    margin: 10px;  
    background: #eee;  
    color: black;  
    padding: 10px;  
}
```

Convirtiéndolos en elemento de bloque (block):



```
<a href="#">One</a>  
<a href="#">Two</a>  
<a href="#">Three</a>  
<a href="#">Four</a>  
<a href="#">Five</a>  
  
a {  
    margin: 10px;  
    background: #eee;  
    color: black;  
    display: block;  
    padding: 10px;  
}
```

En el reverso, supón que tenemos el siguiente menú de navegación:

```
<ul id="nav-menu">  
    <li><a href="#">Home</a></li>  
    <li><a href="#">Dashboard</a></li>  
    <li><a href="#">About</a></li>  
    <li><a href="#">Contact Us</a></li>  
</ul>
```

El HTML normalmente aparecería así:

- [Home](#)
- [Dashboard](#)
- [About](#)
- [Contact Us](#)

Cuando queremos hacer un menú horizontal en la parte superior de nuestra página, podemos hacer lo siguiente:

```
#nav-menu{  
    list-style-type: none; /*This gets rid of the bullets.*/  
}  
#nav-menu li {  
    display: inline;  
}
```

Ahora se ve así:

[Home](#) [Dashboard](#) [About](#) [Contact Us](#)

### Inline-block:

Considera el siguiente ejemplo:

`<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Praesent euismod ultrices ante, ac laoreet nulla vestibulum  
adipiscing. Nam quis justo in augue auctor imperdiet.  
Curabitur aliquet orci sit amet est posuere consectetur. Fusce  
nec leo ut massa viverra venenatis. Nam accumsan libero a  
elit aliquet quis ullamcorper arcu tincidunt. Praesent purus  
turpis, consectetur quis congue vel, pulvinar at lorem.  
Vivamus varius condimentum dolor, quis ultricies ipsum  
porta quis.`

`<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Praesent euismod ultrices ante, ac laoreet nulla vestibulum  
adipiscing. Nam quis justo in augue auctor imperdiet.  
Curabitur aliquet orci sit amet est posuere consectetur. Fusce  
nec leo ut massa viverra venenatis. Nam accumsan libero a  
elit aliquet quis ullamcorper arcu tincidunt. Praesent purus  
turpis, consectetur quis congue vel, pulvinar at lorem.  
Vivamus varius condimentum dolor, quis ultricies ipsum  
porta quis.`

```
p {  
width: 350px; margin: 20px; padding: 20px;  
font-size: 14px;  
background: #eee;  
}
```

Recuerda que las etiquetas `<p>` son elementos block por defecto. Cuando las visualizamos como inline, ambos actúan juntos en un solo párrafo.

`Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent  
euismod ultrices ante, ac laoreet nulla vestibulum adipiscing. Nam quis  
justo in augue auctor imperdiet. Curabitur aliquet orci sit amet est  
posuere consectetur. Fusce nec leo ut massa viverra venenatis. Nam  
accumsan libero a elit aliquet quis ullamcorper arcu tincidunt. Praesent  
purus turpis, consectetur quis congue vel, pulvinar at lorem. Vivamus  
varius condimentum dolor, quis ultricies ipsum porta quis. Lorem  
ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod  
ultrices ante, ac laoreet nulla vestibulum adipiscing. Nam quis justo in  
augue auctor imperdiet. Curabitur aliquet orci sit amet est posuere  
consectetur. Fusce nec leo ut massa viverra venenatis. Nam accumsan  
libero a elit aliquet quis ullamcorper arcu tincidunt. Praesent purus  
turpis, consectetur quis congue vel, pulvinar at lorem. Vivamus varius  
condimentum dolor, quis ultricies ipsum porta quis.`

```
<p>Lorem ipsum dolor sit amet...</p>  
<p>Lorem ipsum dolor sit amet...</p><br>  
p {
```

```
    font-size: 14px;  
    background: #eee;  
    display: inline;  
}
```

Hemos perdido toda capacidad de establecer nuestro ancha y alto puesto que los dos párrafos están completamente integrados.

Los mismos párrafos con inline-block:

*Curabitur aliquet orci sit amet est posuere consectetur. Fusce nec leo ut massa viverra venenatis. Nam accumsan libero a elit aliquet quis ullamcorper arcu tincidunt. Praesent purus turpis, consectetur quis congue vel, pulvinar at lorem. Vivamus varius condimentum dolor, quis ultricies ipsum porta quis.*

*Curabitur aliquet orci sit amet est posuere consectetur. Fusce nec leo ut massa viverra venenatis. Nam accumsan libero a elit aliquet quis ullamcorper arcu tincidunt. Praesent purus turpis, consectetur quis congue vel, pulvinar at lorem. Vivamus varius condimentum dolor, quis ultricies ipsum porta quis.*

```
p {  
    width: 180px;  
    margin: 10px;  
    padding: 20px;  
    font-size: 14px;  
    background: #eee;  
    display: inline-block;  
}
```

Lo que sucede aquí es que le estamos pidiendo al navegador que visualice los párrafos inline, pero permitiéndoles mantener sus características block-level. Esto significa que podemos fijar un ancho y alto manualmente y que ambos elementos permanezcan diferentes, además de aparecer uno junto al otro en el flujo del documento.

## Alineación Vertical

Cuando comenzamos a utilizar inline-block, tal vez habremos notado que la posición vertical de los bloques puede empezar a comportarse de manera diferente a lo que esperábamos, especialmente cuando los bloques tienen diferentes alturas. Por defecto, los elementos se alinean con el borde de más abajo de los bloques dentro de la línea. Para ajustar este comportamiento, podemos usar la propiedad de CSS denominada `vertical-align` la cual tiene valores como top (arriba) y middle (al medio).

### *Default*

**L**orem ipsum dolor sit amet consectetur adipisicing elit. Laudantium rerum provident, perferendis amet quae rem incidunt quaerat sequi, consequuntur sunt distinctio obcaecati facere earum ipsam corrupti! Delectus, laboriosam soluta! Aspernatur!

**Lectura Complementaria:**

<https://uniwebsidad.com/libros/css/capitulo-5?from=librosweb>

# Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding)

\*Puedes descargar un copia del código anterior [acá \(layouts.zip\)](#).

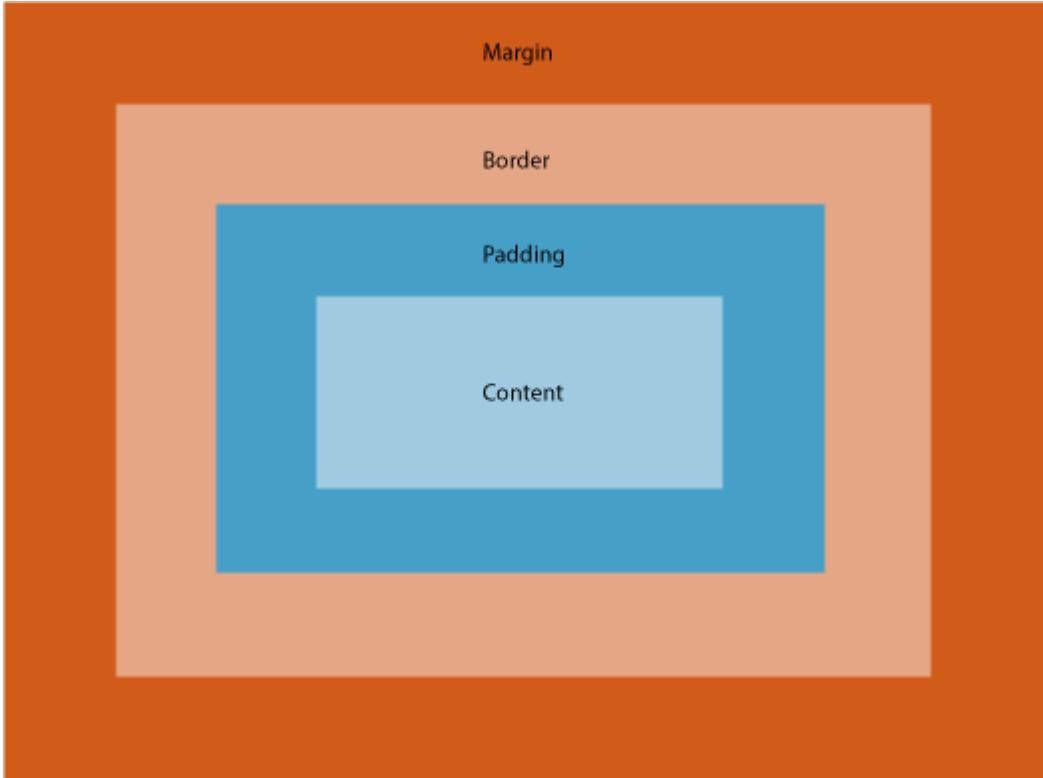
Todos los diseños web se realizan con elementos de bloque o block. Los diseñadores utilizan blocks (normalmente elementos <div>), para crear áreas rectangulares donde caben todos los contenidos. Solo existen 3 reglas:

- **Área total:** El espacio que un elemento ocupa y afecta.
- **Float, clear y overflow**
- **Elementos anidados (nested)**

## Área Total

El ancho total se refiere a cuánto espacio horizontal utiliza un bloque. Esto incluye el margen, el borde y el relleno del bloque. Calcular el ancho, relleno y margen es a menudo el mayor dolor de cabeza para los diseñadores, pero es fácil ver cómo trabajan si utilizas el **modelo de caja**.

El modelo de caja se compone de las propiedades de **margen (margin)**, **borde (border)** y **relleno (padding)**.



El margen está afuera de los elementos de bloque, mientras que el relleno está dentro de ellos. Esto significa que usamos el margen para separar un bloque de las que lo rodean, y relleno para mover el contenido de un bloque lejos de sus bordes.

Podemos definir específicamente el margen, relleno o borde de cualquier lado de un elemento.

```
div {
  padding-top: 10px;
  padding-right: 10px;
  padding-bottom: 10px;
  padding-left: 10px;
}
```

Además puedes usar la propiedad de taquigrafía (shorthand):

```
div {
  padding-top: 25px;
  padding-right: 50px;
  padding-bottom: 75px;
  padding-left: 100px;
}
```

que es equivalente a:

```
div {
  padding: 25px 50px 75px 100px;
```

```
}
```

Y

```
div {  
    padding-top: 25px;  
    padding-right: 50px;  
    padding-bottom: 75px;  
    padding-left: 50px;  
}
```

que es equivalente a:

```
div {  
    padding: 25px 50px 75px;  
}
```

Y

```
div {  
    padding-top: 25px;  
    padding-right: 50px;  
    padding-bottom: 25px;  
    padding-left: 50px;  
}
```

que es equivalente a:

```
div {  
    padding: 25px 50px;  
}
```

Y

```
div {  
    padding-top: 25px;  
    padding-right: 25px;  
    padding-bottom: 25px;  
    padding-left: 25px;  
}
```

que es equivalente a:

```
div{  
    padding: 25px;  
}
```

(El orden va en el sentido del reloj, arriba -> derecha -> abajo -> izquierda).

Entonces, y de acuerdo al modelo de caja, el ancho total de un elemento es:

**(Margin x 2) + (Border x 2) + (Padding x 2) + Content Width (Ancho del contenido)**

Calcular la altura (height) es más complicado ¿Por qué? Porque los márgenes **verticales se desploman**.

Ej:

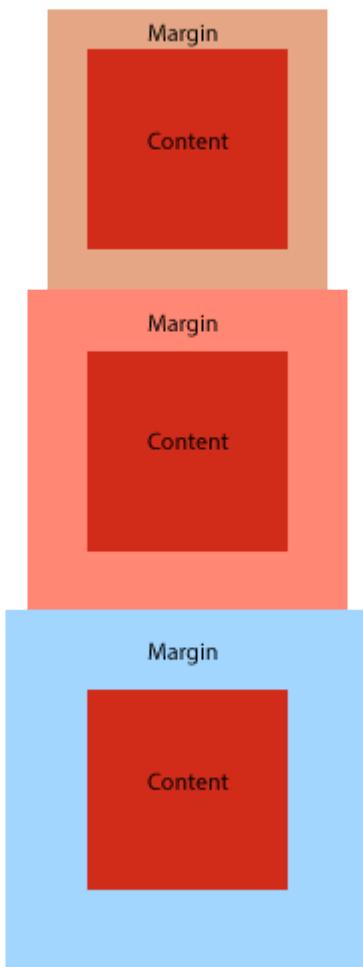
HTML:

```
<div id="box-1">  
</div>  
<div id="box-2">  
</div>  
<div id="box-3">  
</div>
```

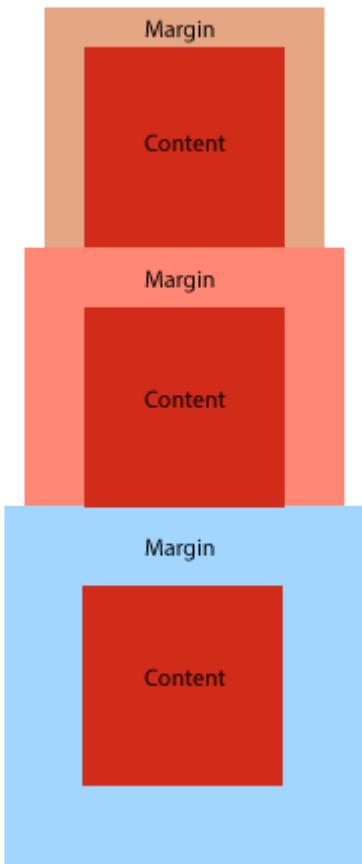
CSS:

```
#box-1, #box-2, #box-3{  
    height: 100px;  
    width: 100px;  
    background-color: red;  
}  
#box-1{  
    margin: 20px;  
}  
#box-2{  
    margin: 30px;  
}  
#box-3{  
    margin: 40px;  
}
```

Tal vez piensas que esto se verá así:



Pero el código se verá más bien así:

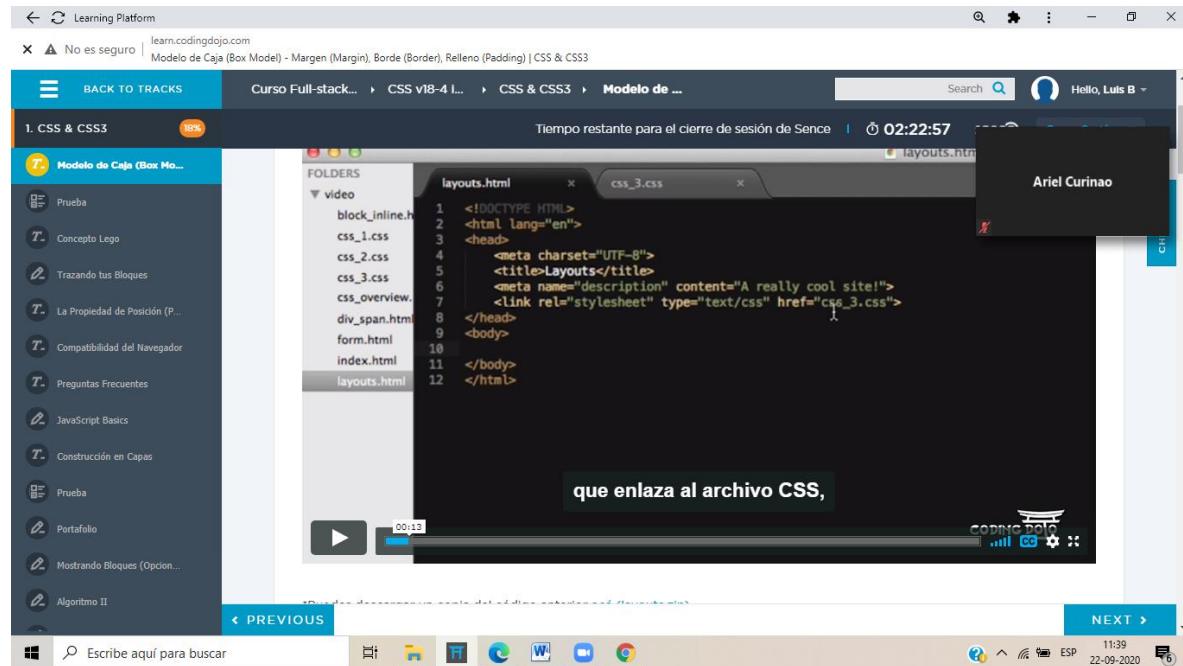


Cuando los márgenes verticales de dos elementos se tocan, **solo el margen del elemento con el valor de margen más grande se mantendrá**, mientras que el margen del elemento con el menor valor de margen se reducirá a cero.

Existen otras situaciones donde los elementos no tienen un derrumbe o reducción en sus márgenes:

- Elementos flotantes
- Elementos absolutamente posicionados
- Elementos inline-block
- Elementos con la propiedad overflow establecida para cualquier otra cosa, menos para que sea visible (Estos no derrumban sus márgenes con sus hijos).
- Elementos eliminados o cleared (Estos no derrumban sus márgenes superiores con el margen inferior de su bloque padre).

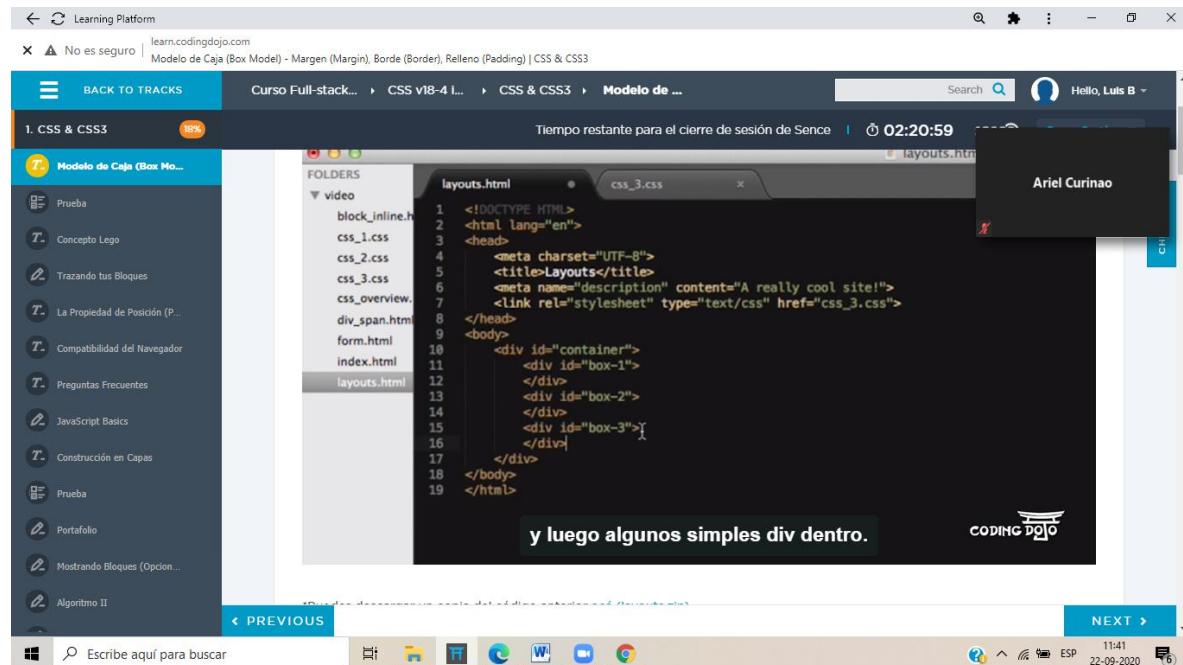
## Archivo layout elazado a css



que enlaza al archivo CSS,

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Layouts</title>
    <meta name="description" content="A really cool site!">
    <link rel="stylesheet" type="text/css" href="css_3.css">
</head>
<body>
</body>
</html>
```

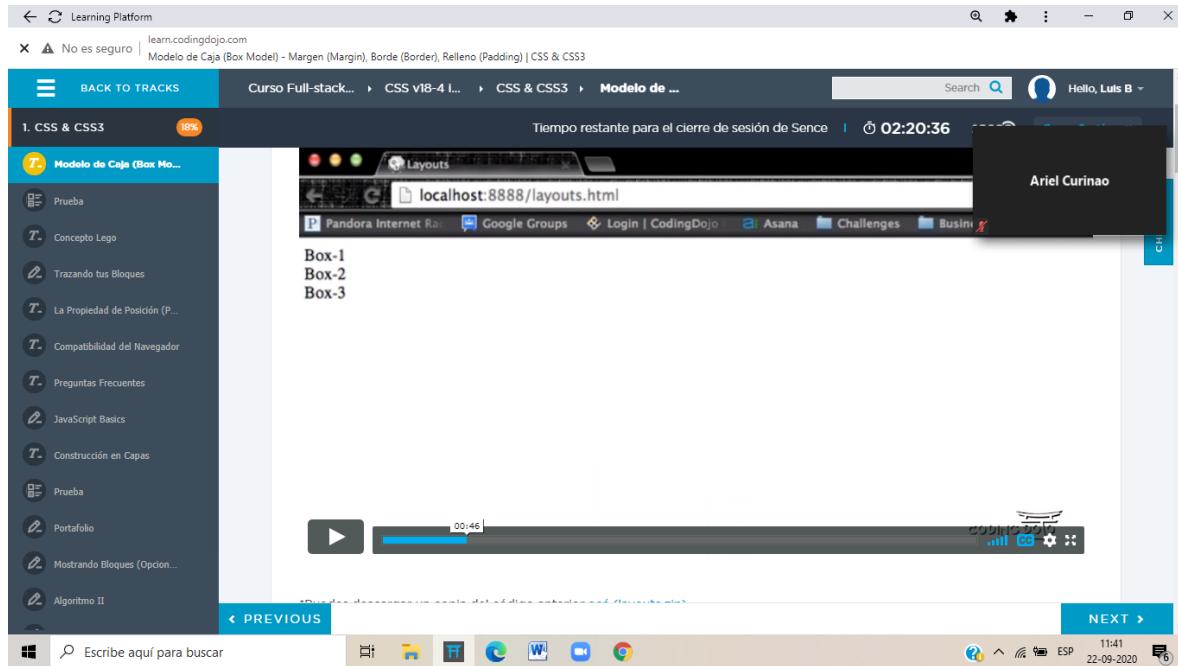
## Cajas con div



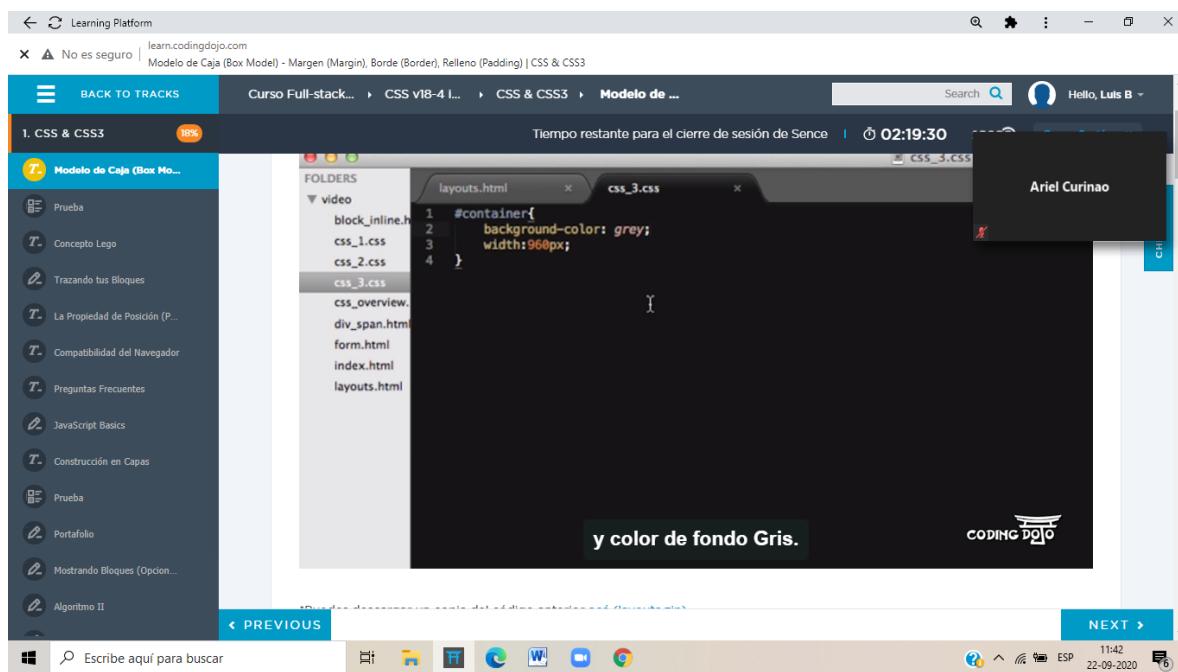
y luego algunos simples div dentro.

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Layouts</title>
    <meta name="description" content="A really cool site!">
    <link rel="stylesheet" type="text/css" href="css_3.css">
</head>
<body>
    <div id="container">
        <div id="box-1">
            <div id="box-2">
                <div id="box-3"></div>
            </div>
        </div>
    </div>
</body>
</html>
```

Asi se ve



Ahora algo de css para el contenedor (container)



Asi se ve gris

Learning Platform

No es seguro | learn.codingdojo.com

Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding) | CSS & CSS3

BACK TO TRACKS

Curso Full-stack... > CSS v18-4 I... > CSS & CSS3 > Modelo de ...

1. CSS & CSS3 18%

T. Modelo de Caja (Box Mo...  
Prueba  
Concepto Lego  
Trazando tus Bloques  
La Propiedad de Posición (P...  
Compatibilidad del Navegador  
Preguntas Frecuentes  
JavaScript Basics  
Construcción en Capas  
Prueba  
Portafolio  
Mostrando Bloques (Opcion...  
Algoritmo II

Time restante para el cierre de sesión de Sence | 02:18:36

Ariel Curinao

localhost:8888/layouts.html

Box-1  
Box-2  
Box-3

Search

Hello, Luis B

NEXT >

Escribe aquí para buscar

## Color dimensiones y borde para las cajas

Learning Platform

No es seguro | learn.codingdojo.com

Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding) | CSS & CSS3

BACK TO TRACKS

Curso Full-stack... > CSS v18-4 I... > CSS & CSS3 > Modelo de ...

1. CSS & CSS3 18%

T. Modelo de Caja (Box Mo...  
Prueba  
Concepto Lego  
Trazando tus Bloques  
La Propiedad de Posición (P...  
Compatibilidad del Navegador  
Preguntas Frecuentes  
JavaScript Basics  
Construcción en Capas  
Prueba  
Portafolio  
Mostrando Bloques (Opcion...  
Algoritmo II

Time restante para el cierre de sesión de Sence | 02:17:47

Ariel Curinao

css\_3.css

```
#container{  
    background-color: grey;  
    width:960px;  
}  
#box-1, #box-2, #box-3{  
    background-color: green;  
    height:200px;  
    width: 200px;  
}
```

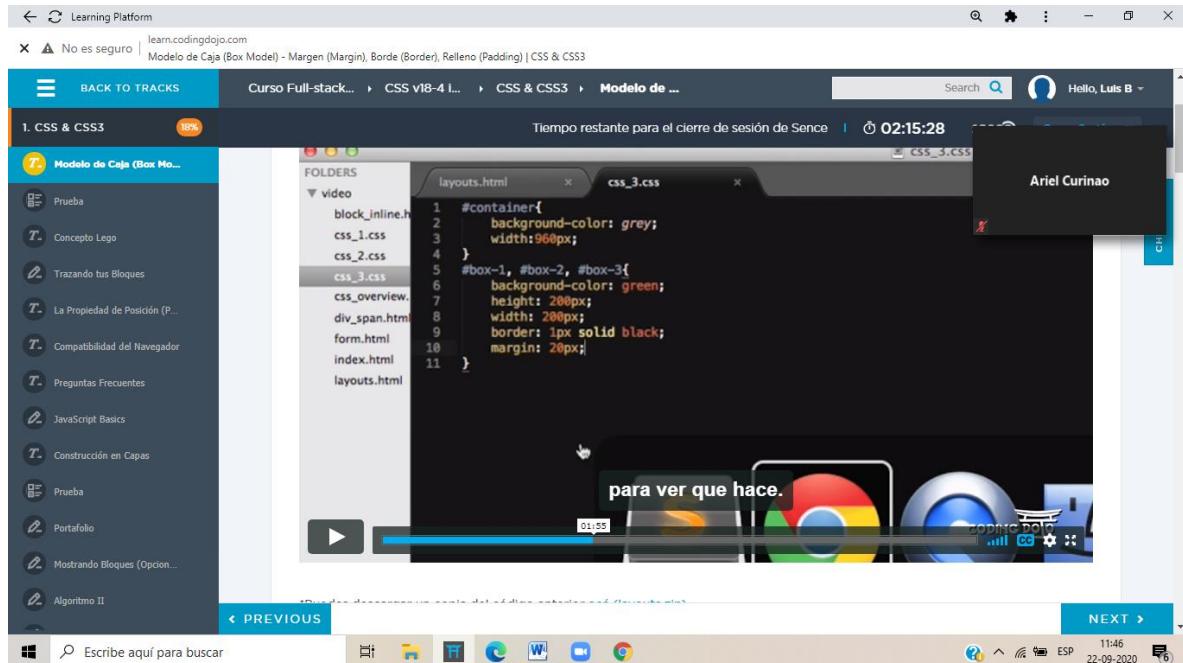
altura y ancho para ellas.

CODING DOJO

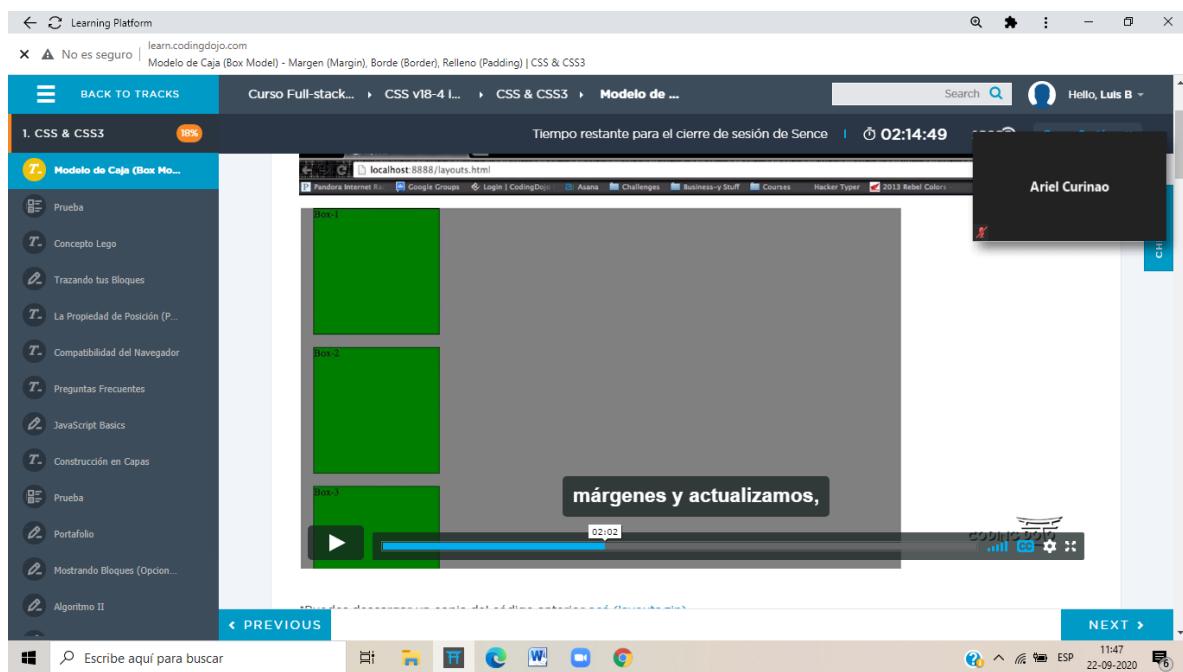
NEXT >

Escribe aquí para buscar

## Margen



ASI SE VEE



AGREGANDO RELLENO (INTERIOR), DIFERENTE AL MARGEN QUE ES EXTERIOR A LAS CAJAS

← ⏪ Learning Platform

X No es seguro | learn.codingdojo.com

Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding) | CSS & CSS3

BACK TO TRACKS

Curso Full-stack... > CSS v18-4 I... > CSS & CSS3 > Modelo de ...

Search Hello, Luis B

1. CSS & CSS3 100%

T. Modelo de Caja (Box Mo...  
Prueba  
Concepto Lego  
Trazando tus Bloques  
La Propiedad de Posición (P...  
Compatibilidad del Navegador  
Preguntas Frecuentes  
JavaScript Basics  
Construcción en Capas  
Prueba  
Portafolio  
Mostrando Bloques (Opcion...  
Algoritmo II

Tiempo restante para el cierre de sesión de Sence | 02:12:08

layouts.html x css\_3.css

FOLDERS  
video  
block\_inline.h...  
css\_1.css  
css\_2.css  
css\_3.css  
css\_overview.h...  
div\_span.html  
form.html  
index.html  
layouts.html

```
#container{  
    background-color: grey;  
    width: 960px;  
}  
  
#box-1, #box-2, #box-3{  
    background-color: green;  
    height: 200px;  
    width: 200px;  
    border: 1px solid black;  
    margin: 20px;  
    padding: 20px;  
}
```

Ariel Curinao

Agreguemos algo de relleno.

CODING DOJO

< PREVIOUS

NEXT >

Escribe aquí para buscar

Windows Search icon

File icon

Home icon

Help icon

Print icon

Search icon

Address bar

11:50 ESP 22-09-2020

ASI SE VEE

Learning Platform | learn.codingdojo.com | No es seguro | Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding) | CSS & CSS3

BACK TO TRACKS | Curso Full-stack... | CSS v18-4 I... | CSS & CSS3 | Modelo de ... | Search | Hello, Luis B | 02:11:37

1. CSS & CSS3 | 100% | Ariel Curinao

Modelo de Caja (Box Model)

- Prueba
- Concepto Lego
- Trazando tus Bloques
- La Propiedad de Posición (Position)
- Compatibilidad del Navegador
- Preguntas Frecuentes
- JavaScript Basics
- Construcción en Capas
- Prueba
- Portafolio
- Mostrando Bloques (Opcionales)
- Algoritmo II

Ahora, para el contenido aquí

NEXT >

Escribe aquí para buscar | 11:50 | 22-09-2020 | PREVIOUS | C

Learning Platform | learn.codingdojo.com | No es seguro | Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding) | CSS & CSS3

BACK TO TRACKS | Curso Full-stack... | CSS v18-4 I... | CSS & CSS3 | Modelo de ... | Search | Hello, Luis B | 02:10:48

1. CSS & CSS3 | 100% | Ariel Curinao

Modelo de Caja (Box Model)

- Prueba
- Concepto Lego
- Trazando tus Bloques
- La Propiedad de Posición (Position)
- Compatibilidad del Navegador
- Preguntas Frecuentes
- JavaScript Basics
- Construcción en Capas
- Prueba
- Portafolio
- Mostrando Bloques (Opcionales)
- Algoritmo II

FOLDERS  
video

layouts.html | css\_3.css

```
#container{  
background-color: grey;  
width: 960px;  
}  
#box-1, #box-2, #box-3{  
background-color: green;  
height: 200px;  
width: 200px;  
border: 1px solid black;  
margin: 20px;  
padding: 20px;  
}
```

Este 0 afectaría a la parte superior e inferior,

NEXT >

Escribe aquí para buscar | 11:51 | 22-09-2020 | PREVIOUS | C

FLOAT MOVER A LA DERECHA EN LINEA CAJA 1

Learning Platform | learn.codingdojo.com | No es seguro | Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding) | CSS & CSS3

BACK TO TRACKS | Curso Full-stack... | CSS v18-4 I... | CSS & CSS3 | Modelo de ... | Search | Hello, Luis B | 02:09:09

1. CSS & CSS3 | 100% | Ariel Curinao

Modelo de Caja (Box Model)

- Prueba
- Concepto Lego
- Trazando tus Bloques
- La Propiedad de Posición (Position)
- Compatibilidad del Navegador
- Preguntas Frecuentes
- JavaScript Basics
- Construcción en Capas
- Prueba
- Portafolio
- Mostrando Bloques (Opcion...
- Algoritmo II

FOLDERS | layouts.html | css\_3.css

```
#container{ background-color: grey; width:960px; }#box-1, #box-2, #box-3{ background-color: green; height: 200px; width: 200px; border: 1px solid black; margin: 0 20px; padding: 20px; }#box-1{ float:right; }
```

Como ya está a la izquierda, voy a hacer float a la derecha.

NEXT >

Escribe aquí para buscar | 11:53 | 22-09-2020

Learning Platform | learn.codingdojo.com | No es seguro | Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding) | CSS & CSS3

BACK TO TRACKS | Curso Full-stack... | CSS v18-4 I... | CSS & CSS3 | Modelo de ... | Search | Hello, Luis B | 02:08:18

1. CSS & CSS3 | 100% | Ariel Curinao

Modelo de Caja (Box Model)

- Prueba
- Concepto Lego
- Trazando tus Bloques
- La Propiedad de Posición (Position)
- Compatibilidad del Navegador
- Preguntas Frecuentes
- JavaScript Basics
- Construcción en Capas
- Prueba
- Portafolio
- Mostrando Bloques (Opcion...
- Algoritmo II

localhost:8888/layouts.html

Ahora, como puedes ver, la caja 1

NEXT >

Escribe aquí para buscar | 11:54 | 22-09-2020

FLOAT A TODAS

A screenshot of a video player interface. The video is titled "Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding) | CSS & CSS3". The video progress bar shows 03:55. A text overlay on the video says "Y si colocamos float a todas,". The video player has a dark theme with a play button and a progress bar.

Code snippet shown in the video:

```
#container{  
    background-color: grey;  
    width: 500px;  
}  
#box-1, #box-2, #box-3{  
    background-color: green;  
    width: 200px;  
    border: 1px solid black;  
    margin: 10px;  
    padding: 10px;  
}  
#box-1, #box-2, #box-3{  
    float: right;  
}
```

A screenshot of a web browser window showing three green rectangular boxes labeled "Box-3", "Box-2", and "Box-1" arranged horizontally. The browser toolbar at the top includes icons for file, edit, view, insert, format, table, form, and help. The status bar at the bottom shows "11:54 22-09-2020".

Text overlay on the video says "están todas juntas,". The video player has a dark theme with a play button and a progress bar.

TODAS JUNTAS PERO GRIS DESAPARECIO, CONTENEDOR COLAPSA

DOS FORMAS DE EVITAR

OVERFLOW

← ⏪ Learning Platform

X No es seguro | learn.codingdojo.com

Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding) | CSS & CSS3

BACK TO TRACKS

Curso Full-stack... > CSS v18-4 I... > CSS & CSS3 > Modelo de ...

Search Hello, Luis B

Tiempo restante para el cierre de sesión de Sence 02:05:58

1. CSS & CSS3 100%

Modelo de Caja (Box Model)

FOLDERS

video

block\_inline.html  
css\_1.css  
css\_2.css  
**css\_3.css**

```
#container{  
    background-color: grey;  
    width: 960px;  
    overflow: hidden;  
    height: auto;  
}  
  
#box-1, #box-2, #box-3{  
    background-color: green;  
    height: 200px;  
    width: 200px;  
    border: 1px solid black;  
    margin: 0 20px;  
    padding: 20px;  
}  
  
#box-1, #box-2, #box-3{  
    float: right;  
}
```

Ariel Curinao

y height auto.

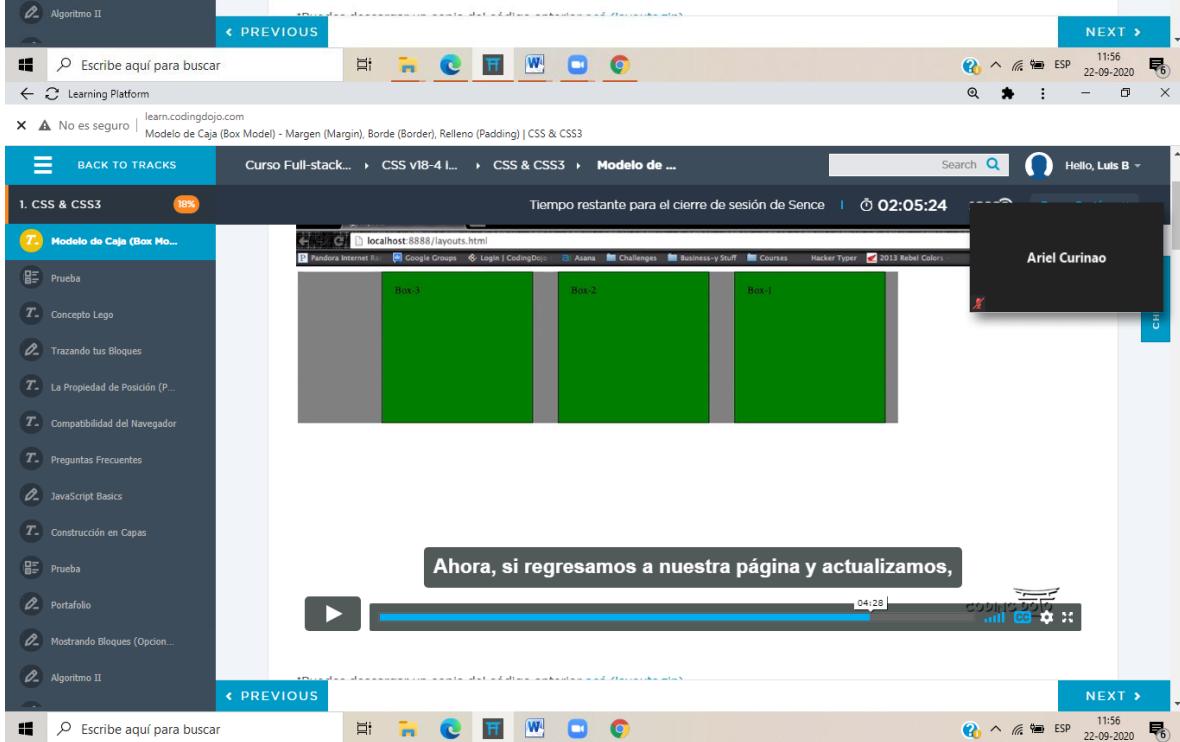
CODING DOJO

< PREVIOUS NEXT >

Escribe aquí para buscar

Learning Platform

11:56 22-09-2020



## O DARLE CLASE EN LAYOUT

The screenshot shows a web-based learning environment for 'Curso Full-stack... > CSS v18-4 I... > CSS & CSS3 > Modelo de ...'. The main area displays a code editor with two files: 'layouts.html' and 'css\_3.css'. The 'layouts.html' file contains:

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Layouts</title>
    <meta name="description" content="A really cool site!">
    <link rel="stylesheet" type="text/css" href="css_3.css">
</head>
<body>
    <div id="container">
        <div id="box-1">Box-1</div>
        <div id="box-2">Box-2</div>
        <div id="box-3">Box-3</div>
    </div>
    <div class="clear"></div>
</body>
</html>
```

The 'css\_3.css' file contains:

```
body { margin: 0; padding: 0; }
#container { width: 100%; height: 100%; border: 1px solid black; position: relative; }
#box-1, #box-2, #box-3 { width: 33%; height: 100%; border: 1px solid black; position: absolute; top: 0; left: 0; }
.clear { clear: both; }
```

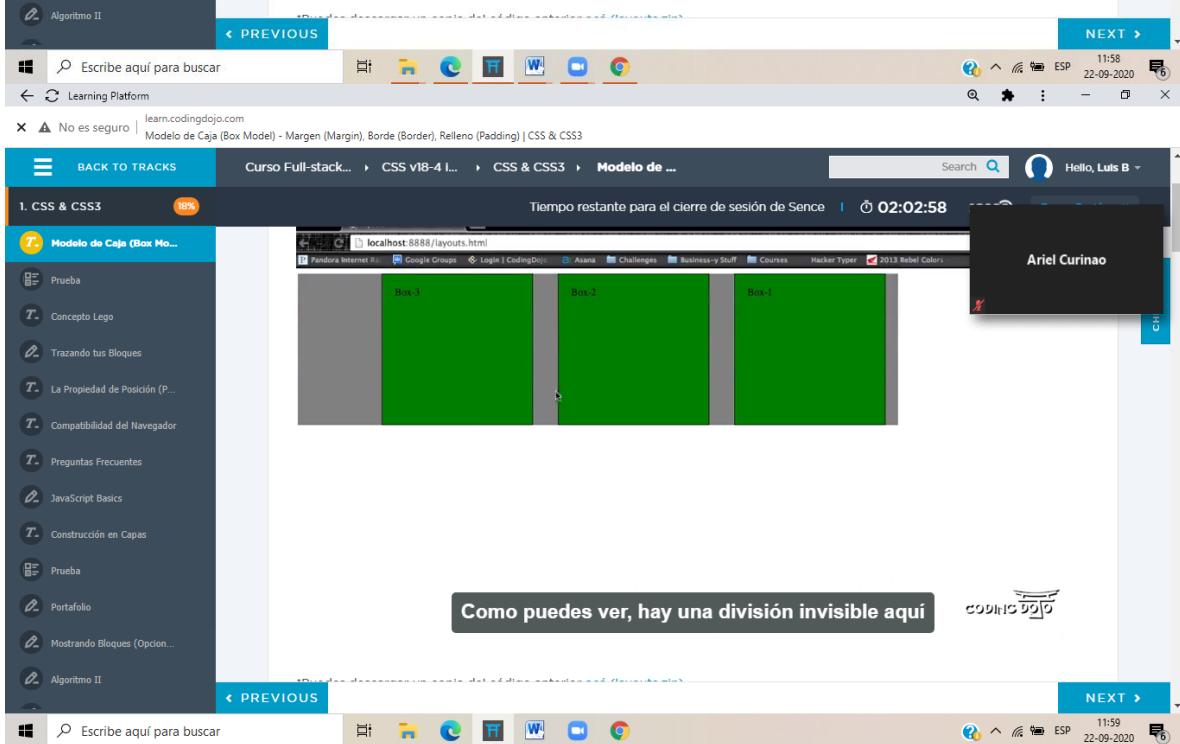
A tooltip is overlaid on the 'clear' class in the CSS code, containing the text: "Solo voy a darle la clase clear,".

Y EN CSS APlicar CLEAR

A screenshot of a web-based learning platform. The top navigation bar shows 'BACK TO TRACKS', 'Curso Full-stack...', 'CSS v18-4 I...', 'CSS & CSS3', and 'Modelo de ...'. A timer indicates 'Tiempo restante para el cierre de sesión de Sence' (Time remaining for the session to end) at 02:03:31. The main content area has a sidebar with a list of topics under '1. CSS & CSS3'. The main panel displays a code editor for 'css\_3.css' with the following CSS:

```
#container{background-color: grey; width:960px; overflow: hidden; height: auto;}#box-1, #box-2, #box-3{background-color: green; height: 200px; width: 200px; border: 1px solid black; margin: 0 20px; padding: 20px;}#box-1, #box-2, #box-3{float:right;}
```

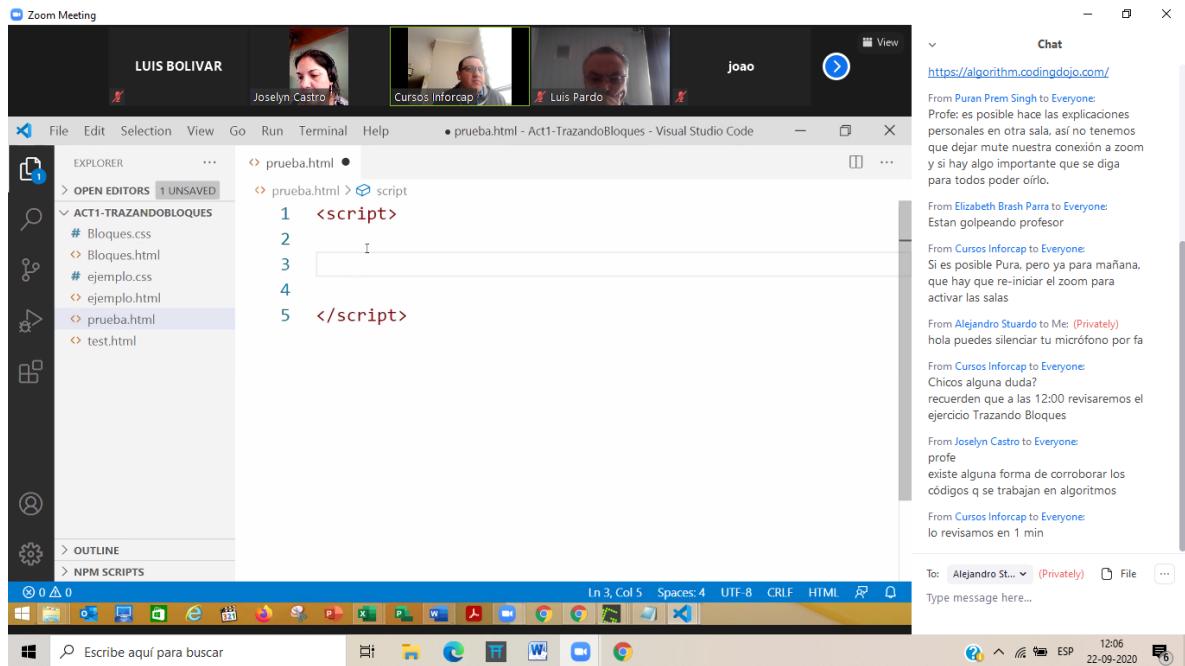
A tooltip box highlights the 'clear:both' property with the text: 'para la clase clear, aplico la propiedad clear: both.'



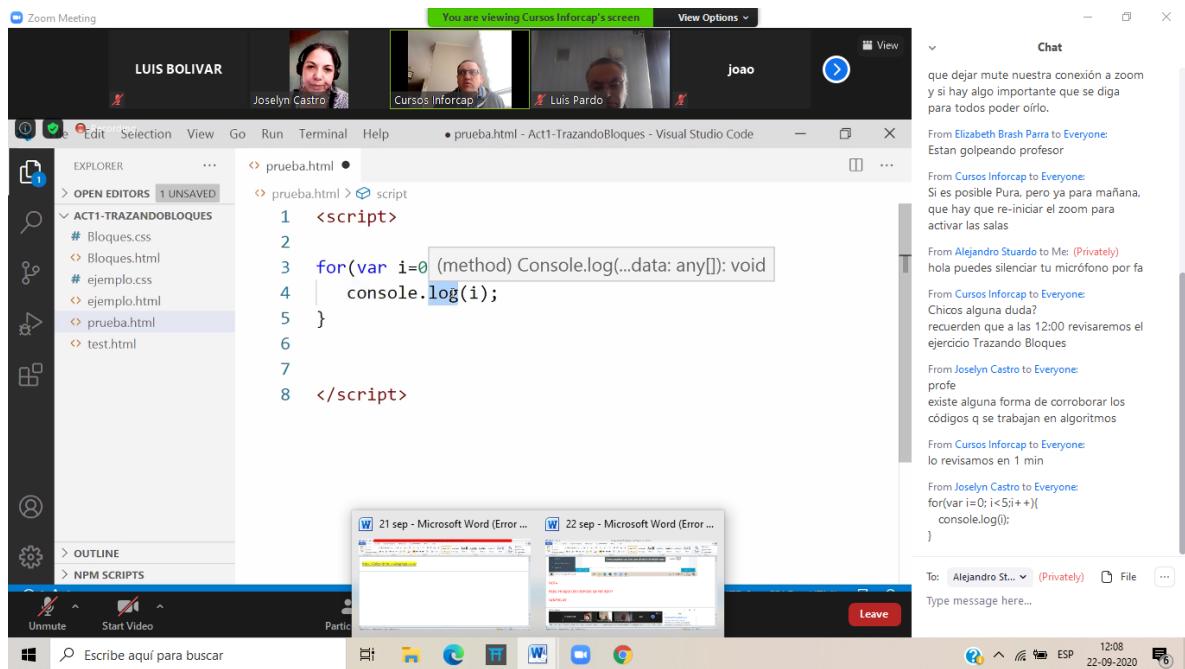
## NOTA

PARA PROBAR LOS CODIGOS EN PRE BOOT

DENTRO DE SCRIPT



## EJEMPLO



A screenshot of a Zoom meeting interface within Visual Studio Code. The top bar shows the Zoom logo and participant names: LUIS BOLIVAR, Joselyn Castro, Luis Pardo, and Alejandro Stuardo. The main area is a code editor with an HTML file named 'prueba.html'. The code contains a script block with a for loop that logs the numbers 0 to 5 to the console.

```
<script>
for(var i=0; i<6; i++){
    console.log(i);
}
</script>
```

ESTE ES CON 6 HAY QUE EJECUTAR LA PAG Y PRECIONAR F12

## OTRO EJ

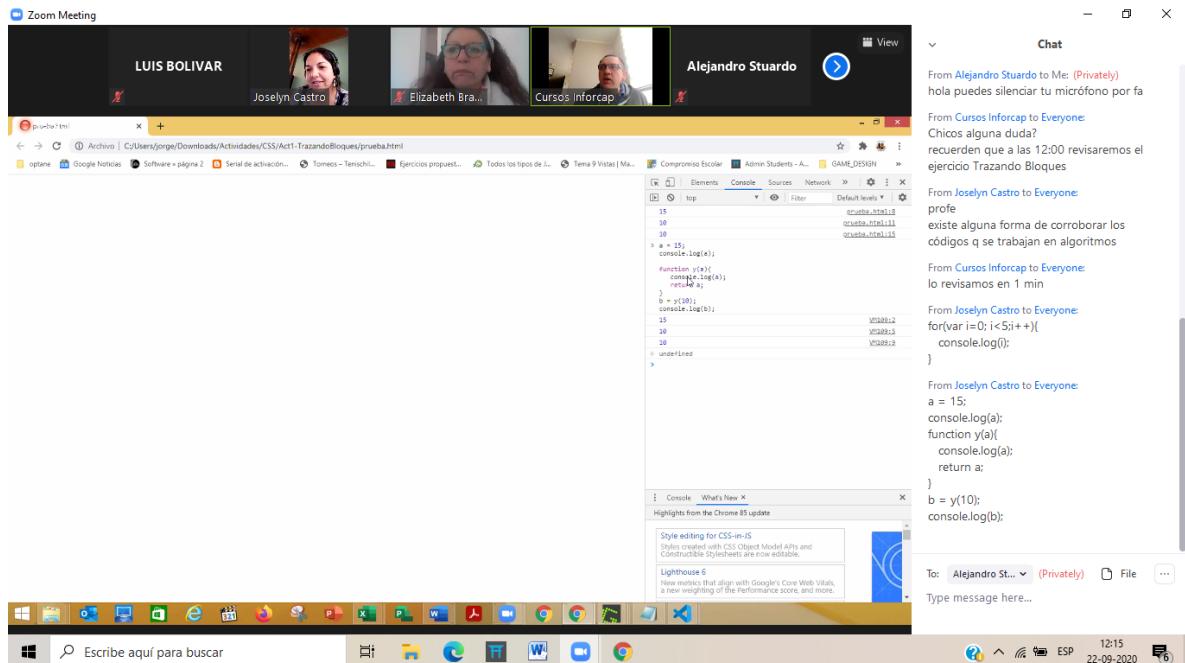
A screenshot of a Zoom meeting interface within Visual Studio Code. The top bar shows the Zoom logo and participant names: LUIS BOLIVAR, Joselyn Castro, Elizabeth Bra..., Cursos Inforcap, and Alejandro Stuardo. The main area is a code editor with an HTML file named 'prueba.html'. The code contains a script block with a function 'y' that logs the value of 'a' to the console and returns it. It then calls this function with the argument 10 and logs the result.

```
a = 15;
console.log(a);

function y(a){
    console.log(a);
    return a;
}

b = y(10);
console.log(b);
```

TAMBIEN SE PUEDE PROBAR EN UNA CONSOLA CUALQUIERA



CONTINUACION AHORA

# Modelo de Caja (Box Model) - Margen (Margin), Borde (Border), Relleno (Padding)

\*Puedes descargar un copia del código anterior [acá \(layouts.zip\)](#).

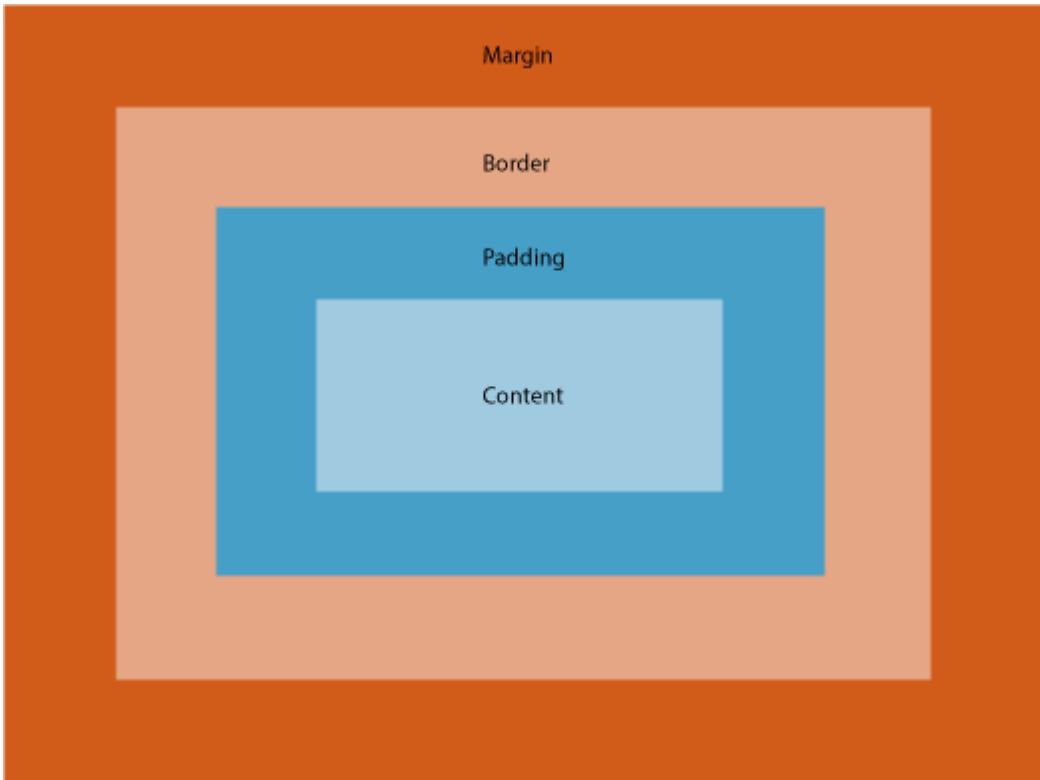
Todos los diseños web se realizan con elementos de bloque o block. Los diseñadores utilizan blocks (normalmente elementos <div>), para crear áreas rectangulares donde caben todos los contenidos. Solo existen 3 reglas:

- **Área total:** El espacio que un elemento ocupa y afecta.
- **Float, clear y overflow**
- **Elementos anidados (nested)**

## Área Total

El ancho total se refiere a cuánto espacio horizontal utiliza un bloque. Esto incluye el margen, el borde y el relleno del bloque. Calcular el ancho, relleno y margen es a menudo el mayor dolor de cabeza para los diseñadores, pero es fácil ver cómo trabajan si utilizas el **modelo de caja**.

El modelo de caja se compone de las propiedades de **margen (margin)**, **borde (border)** y **relleno (padding)**.



El margen está afuera de los elementos de bloque, mientras que el relleno está dentro de ellos. Esto significa que usamos el margen para separar un bloque de los que lo rodean, y relleno para mover el contenido de un bloque lejos de sus bordes.

Podemos definir específicamente el margen, relleno o borde de cualquier lado de un elemento.

```
div {  
  padding-top: 10px;  
  padding-right: 10px;  
  padding-bottom: 10px;  
  padding-left: 10px;  
}
```

Además puedes usar la propiedad de taquigrafía (shorthand):

```
div {  
  padding: 25px 50px 75px 100px;  
}
```

que es equivalente a:

```
div {  
    padding: 25px 50px 75px 100px;  
}
```

Y

```
div {  
    padding-top: 25px;  
    padding-right: 50px;  
    padding-bottom: 75px;  
    padding-left: 100px;  
}
```

que es equivalente a:

```
div {  
    padding: 25px 50px 75px;  
}
```

Y

```
div {  
    padding-top: 25px;  
    padding-right: 50px;  
    padding-bottom: 25px;  
    padding-left: 50px;  
}
```

que es equivalente a:

```
div {  
    padding: 25px 50px;  
}
```

Y

```
div {  
    padding-top: 25px;  
    padding-right: 25px;  
    padding-bottom: 25px;  
    padding-left: 25px;  
}
```

que es equivalente a:

```
div{  
    padding: 25px;  
}
```

(El orden va en el sentido del reloj, arriba -> derecha -> abajo -> izquierda).

Entonces, y de acuerdo al modelo de caja, el ancho total de un elemento es:

**(Margin x 2) + (Border x 2) + (Padding x 2) + Content Width (Ancho del contenido)**

Calcular la altura (height) es más complicado ¿Por qué? Porque los márgenes **verticales se desploman**.

Ej:

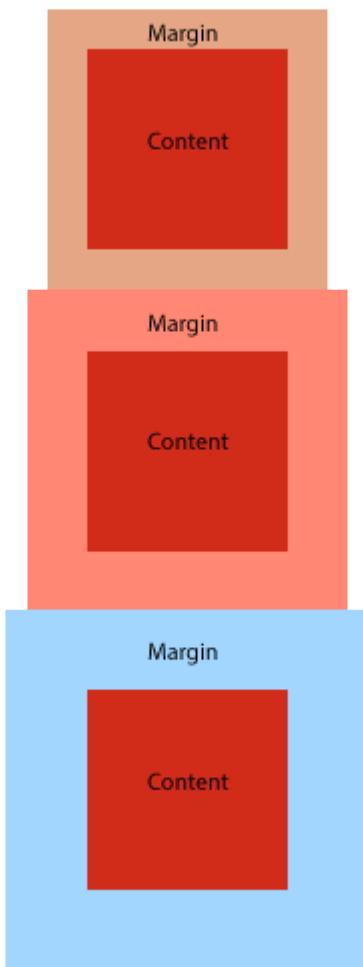
HTML:

```
<div id="box-1">  
</div>  
<div id="box-2">  
</div>  
<div id="box-3">  
</div>
```

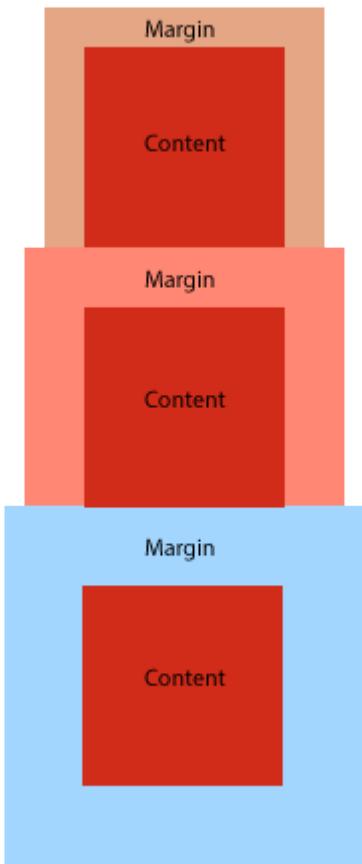
CSS:

```
#box-1, #box-2, #box-3{  
    height: 100px;  
    width: 100px;  
    background-color: red;  
}  
#box-1{  
    margin: 20px;  
}  
#box-2{  
    margin: 30px;  
}  
#box-3{  
    margin: 40px;  
}
```

Tal vez piensas que esto se verá así:



Pero el código se verá más bien así:



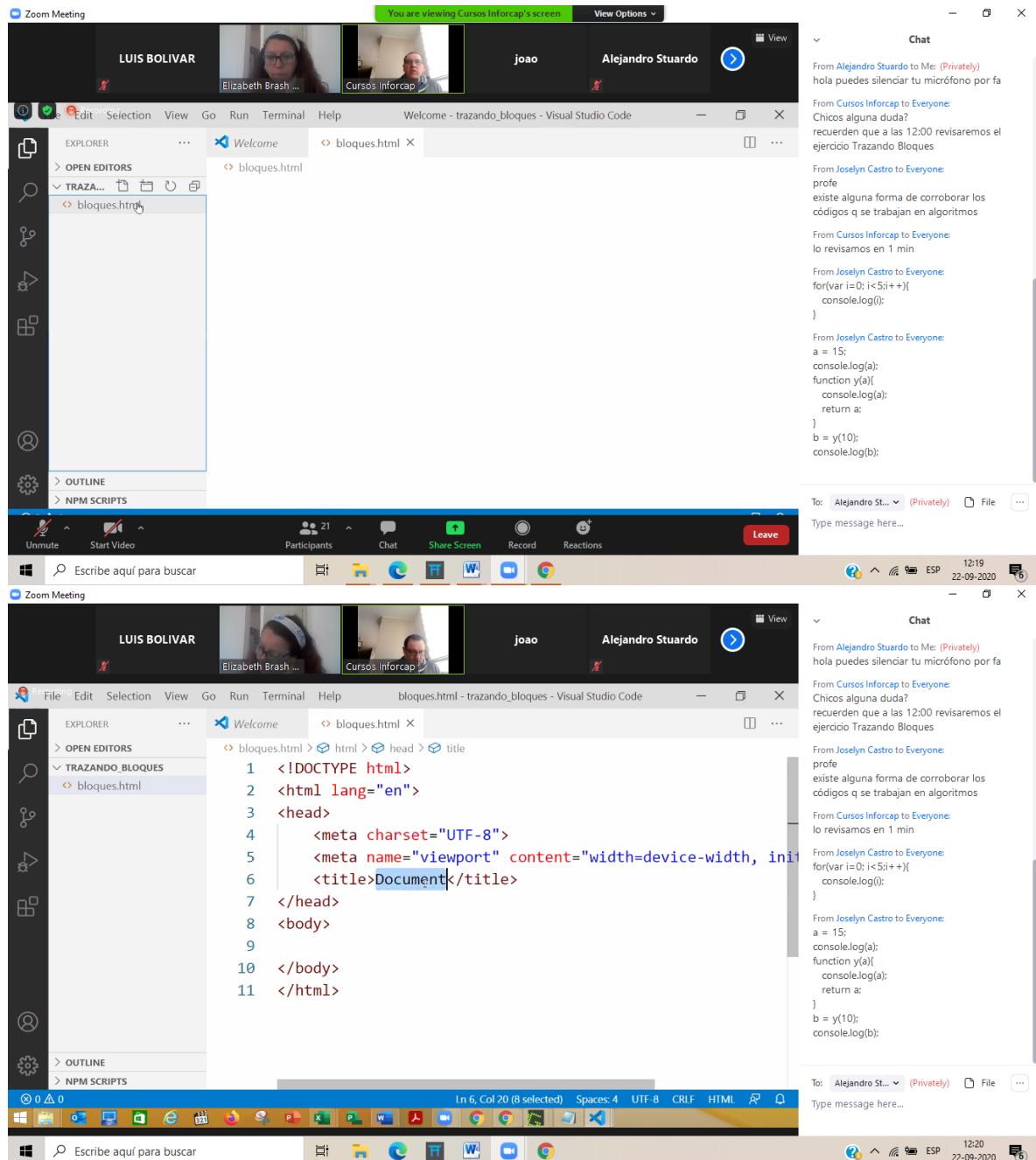
Cuando los márgenes verticales de dos elementos se tocan, **solo el margen del elemento con el valor de margen más grande se mantendrá**, mientras que el margen del elemento con el menor valor de margen se reducirá a cero.

Existen otras situaciones donde los elementos no tienen un derrumbe o reducción en sus márgenes:

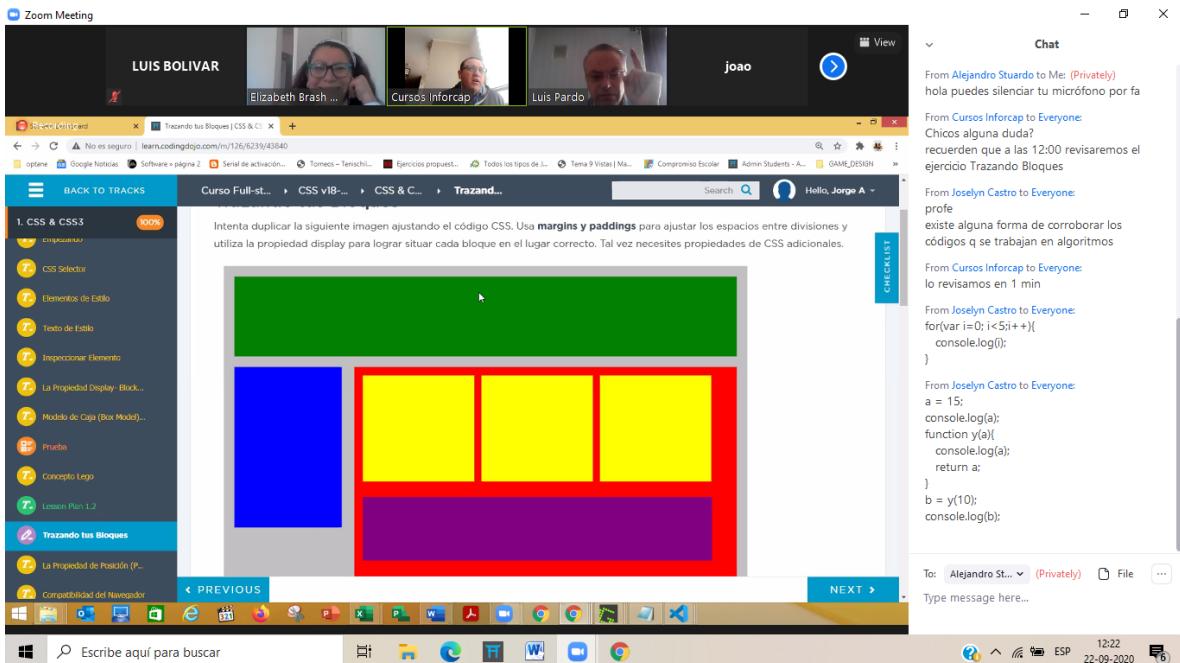
- Elementos flotantes
- Elementos absolutamente posicionados
- Elementos inline-block
- Elementos con la propiedad overflow establecida para cualquier otra cosa, menos para que sea visible (Estos no derrumban sus márgenes con sus hijos).
- Elementos eliminados o cleared (Estos no derrumban sus márgenes superiores con el margen inferior de su bloque padre).

## TAREA DE TRAZANDO BLOQUE

### CREAMOS LA PAG



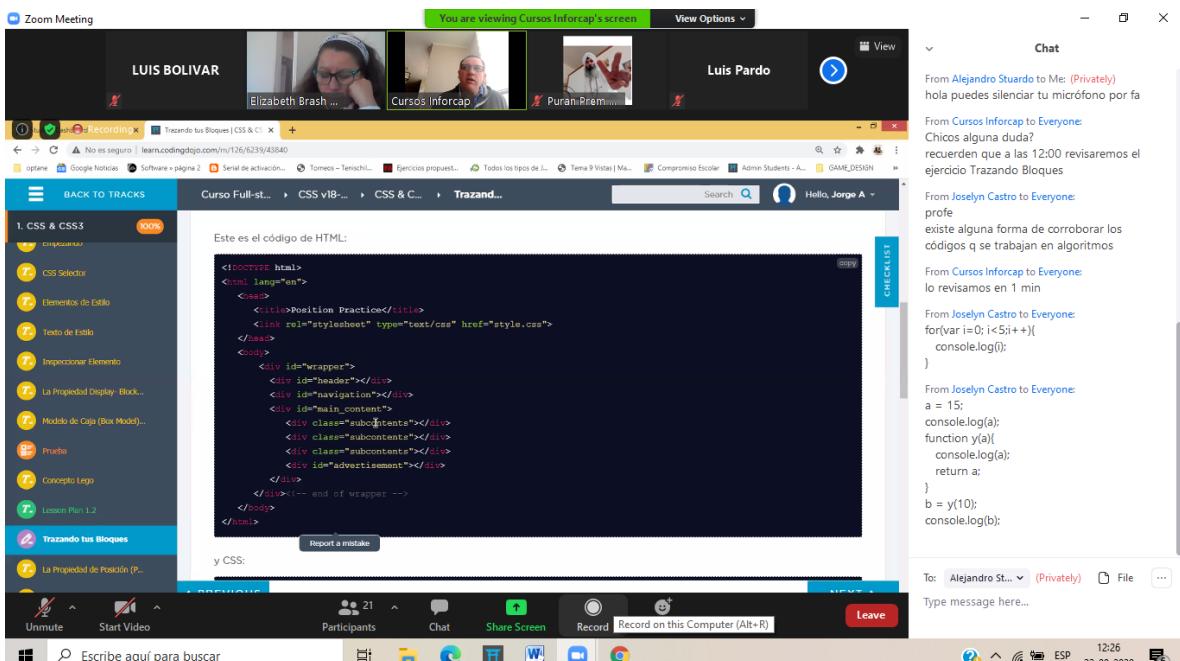
ESTO ES LO QUE SE PRETENDE HACER



## VER LOS VIDEOS (DESCRIPCION DE WHATSAPP)

## VER LA ESTRUCTURA

## ENCABEZADO, CUARTO DEL LADO, BLOG GENERAL CON CUADROS Y BASE..

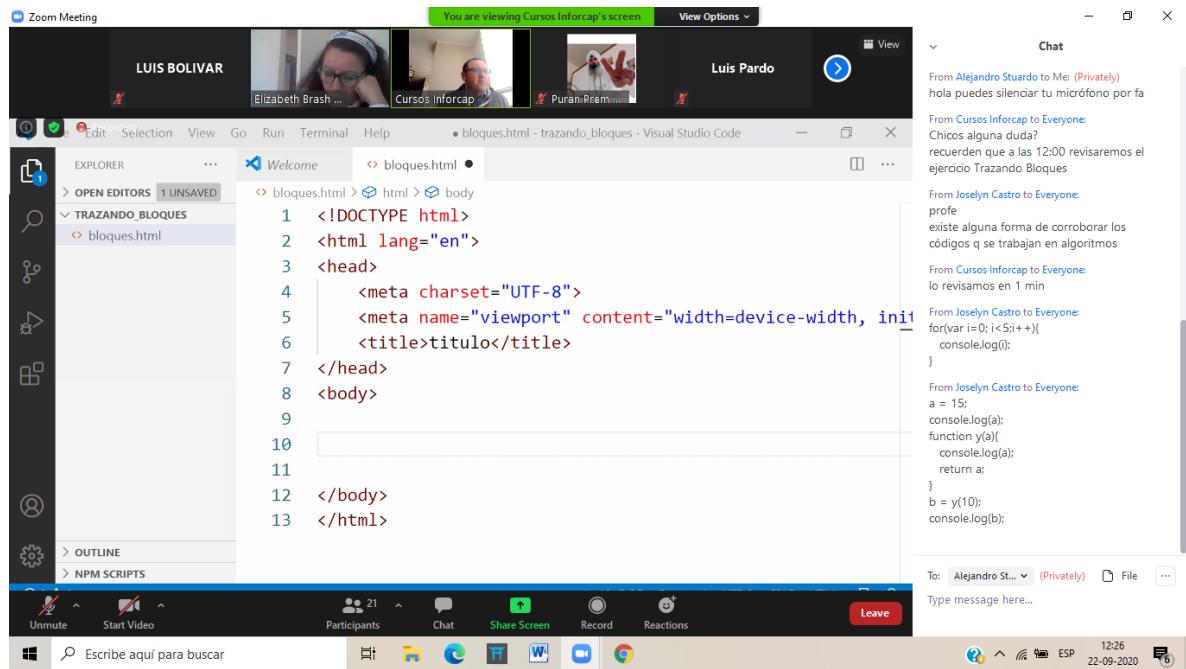


## NOS DA EL CODIGO BASE

PERO VAMOS DE APOCO

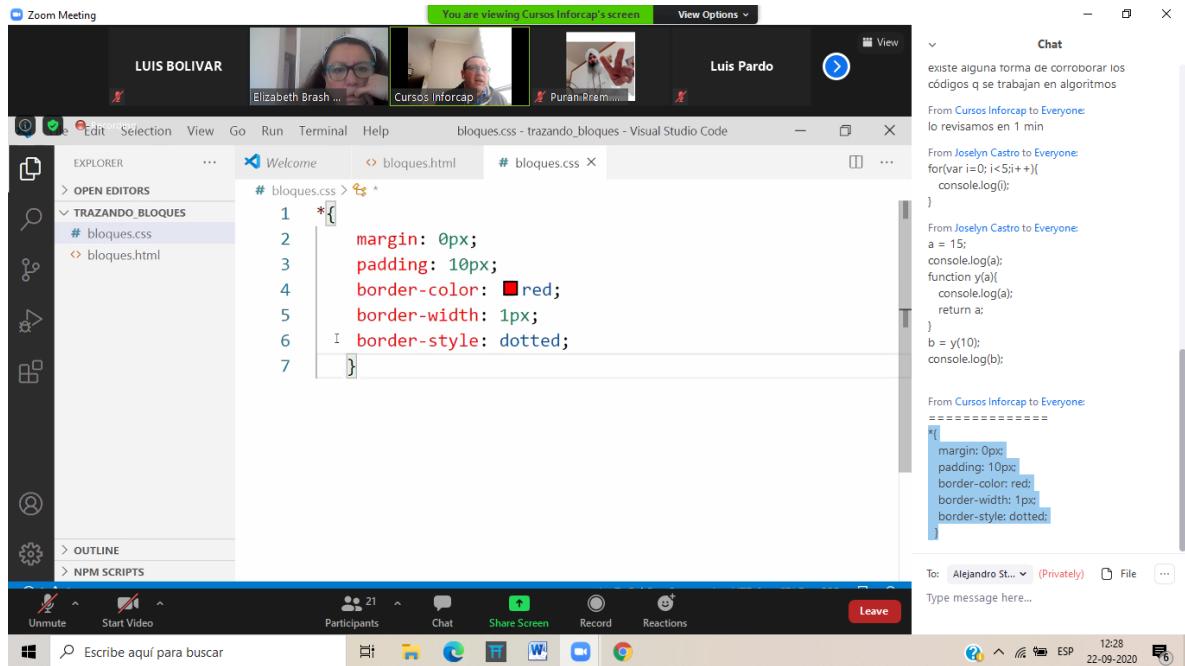
VAMOS A DENTRO DEL BODY

## VISUALIZAR LAS CAJAS



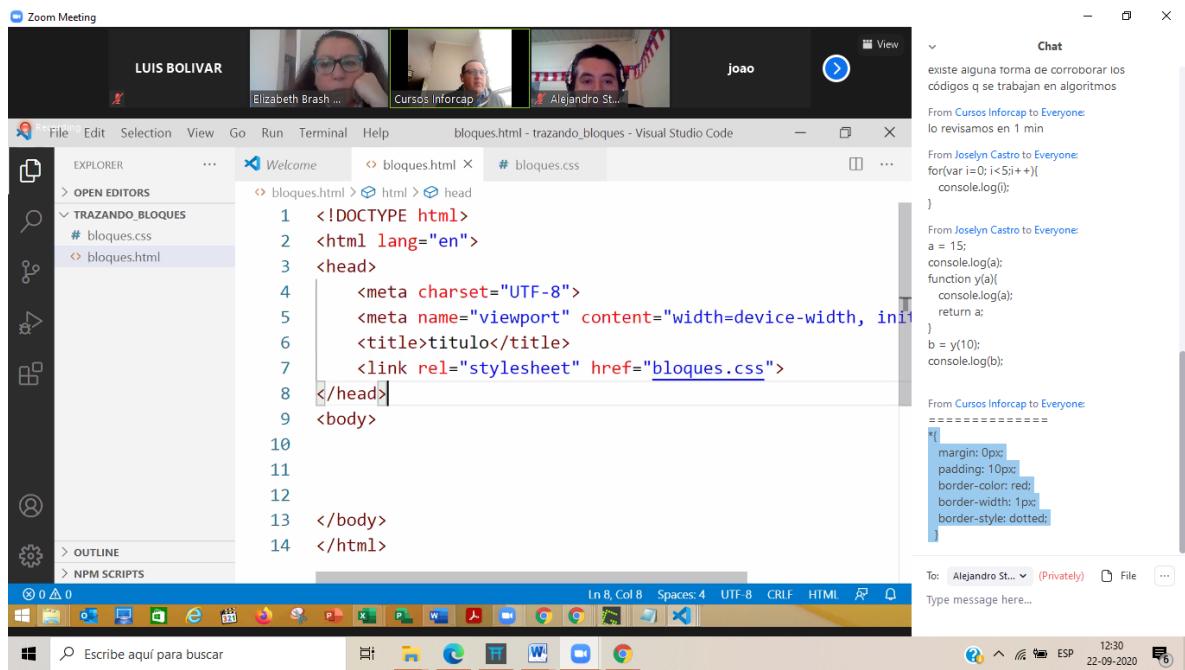
## CODIGO MAGICO PARA ENTRADA CSS

```
*{
    margin: 0px;
    padding: 10px;
    border-color: red;
    border-width: 1px;
    border-style: dotted;
}
```

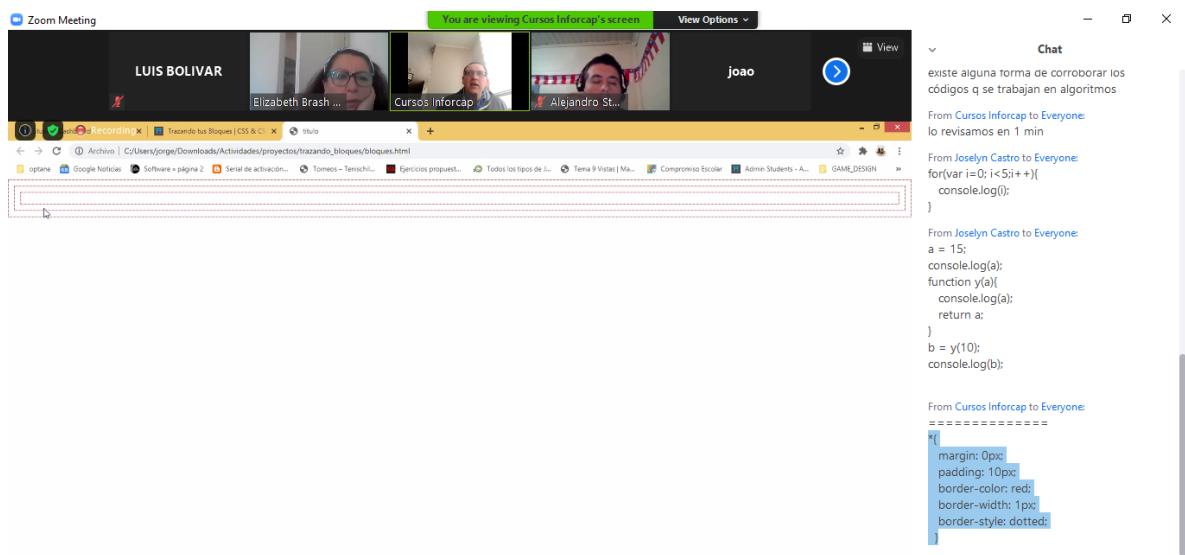
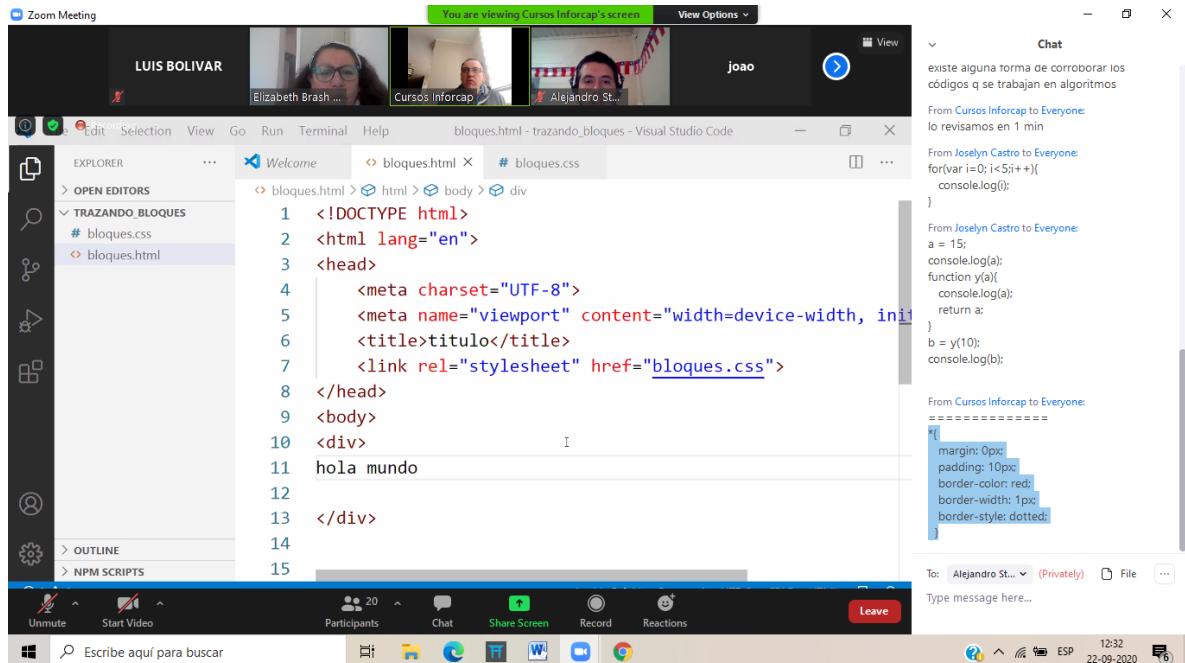


## ENLACE CON CSS

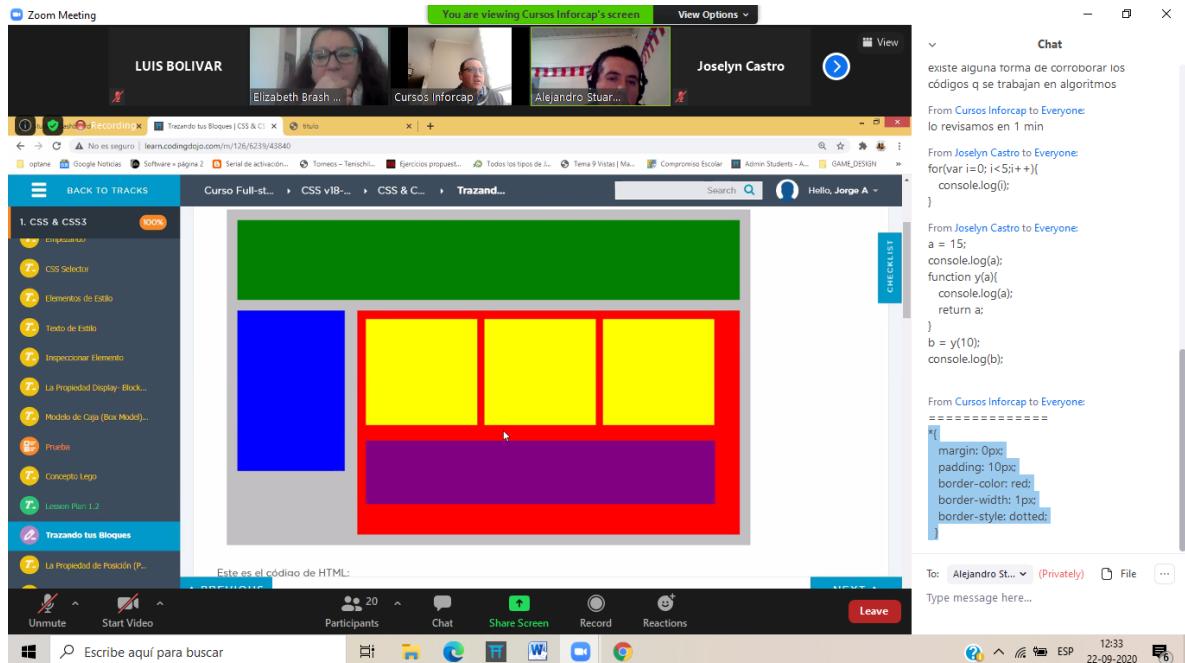
### EL BIOQUE CSS DEBE ESTAR AL lado DE LA PAG Y REALIZAR LA LINEA LINK (7)



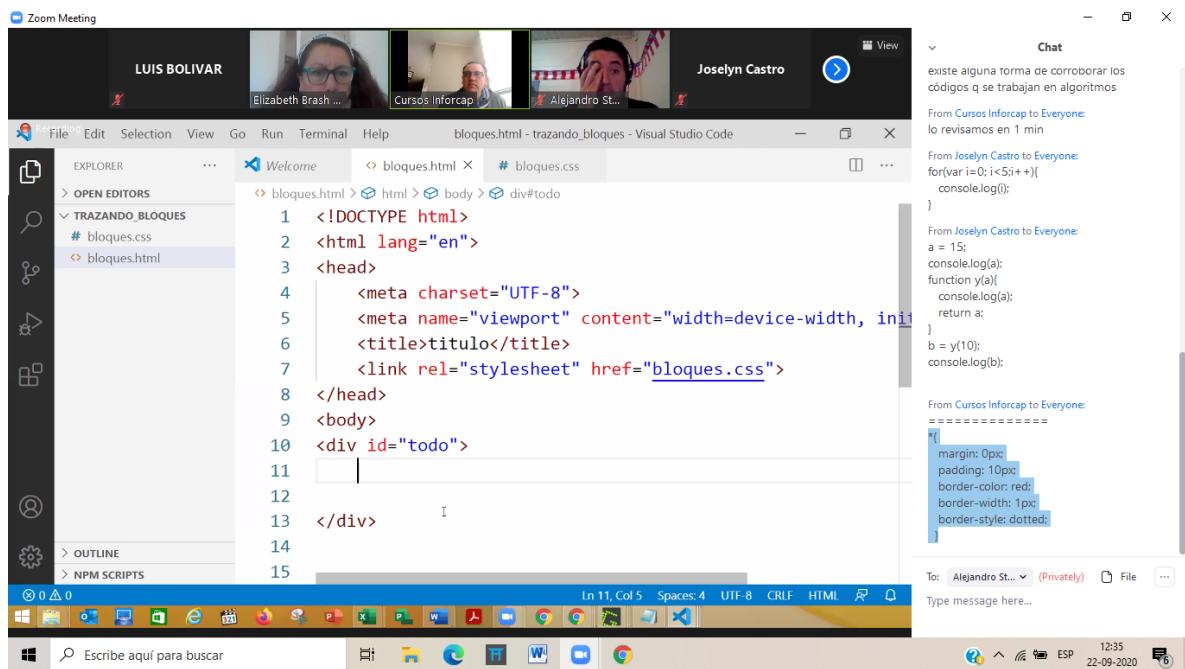
## HAGAMOS LAS CAJAS



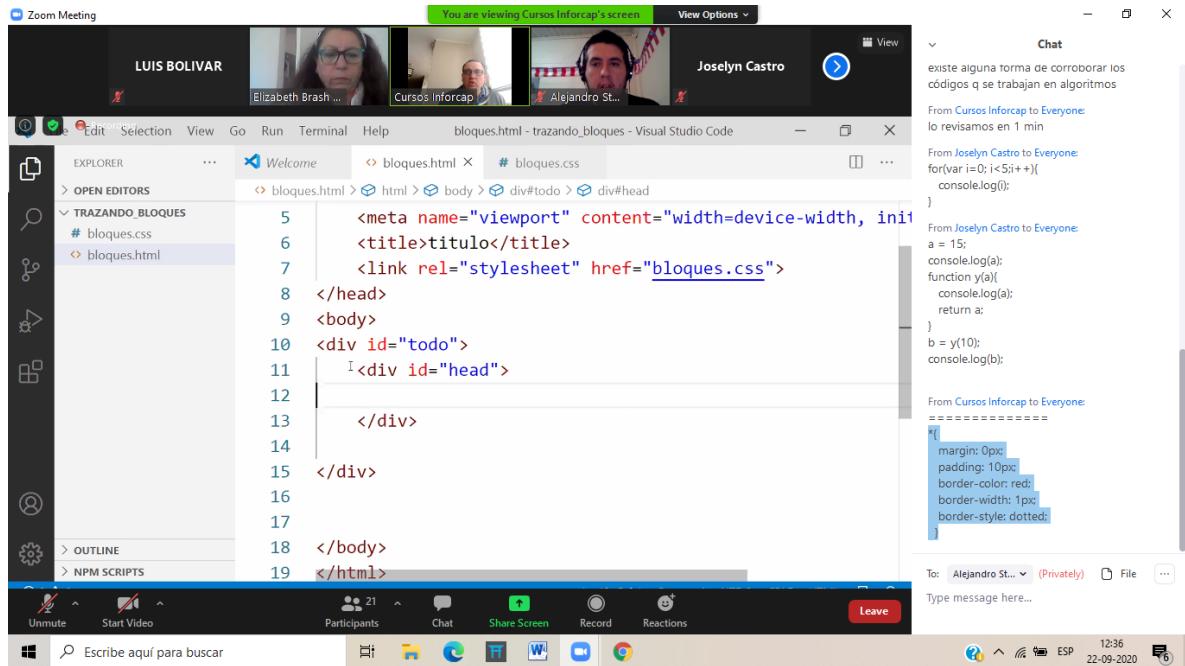
HAY 8 DIV



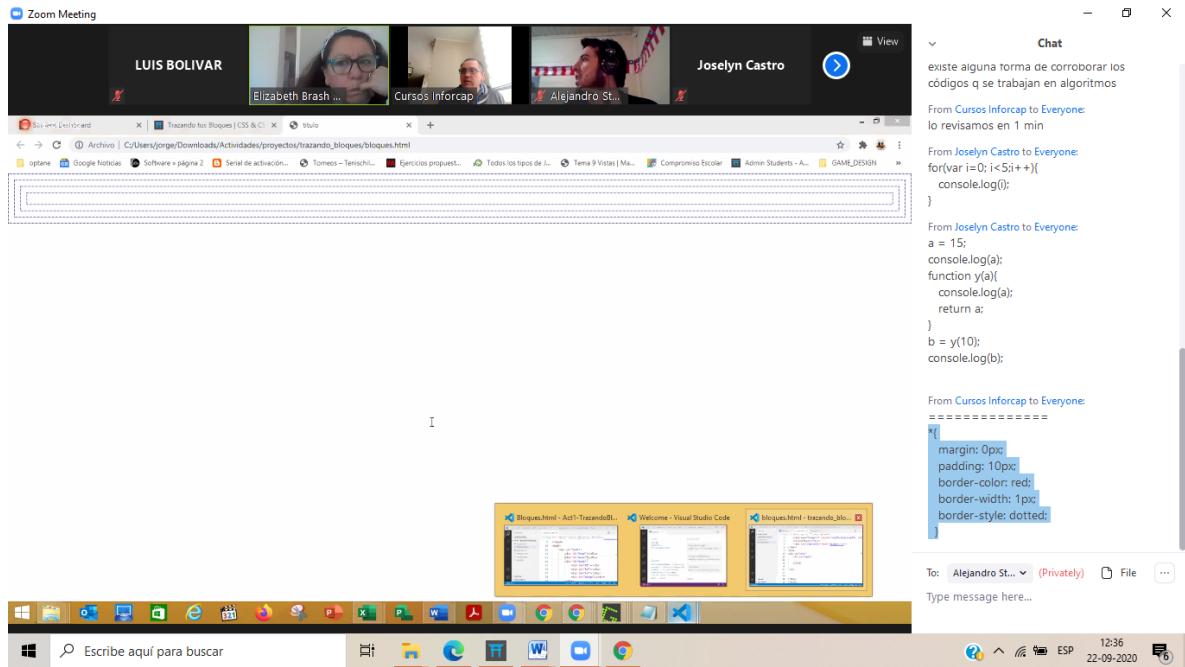
## PRIMERO EL DIV CON ID TODO QUE PUEDE SER CUALQUIER NOMBRE



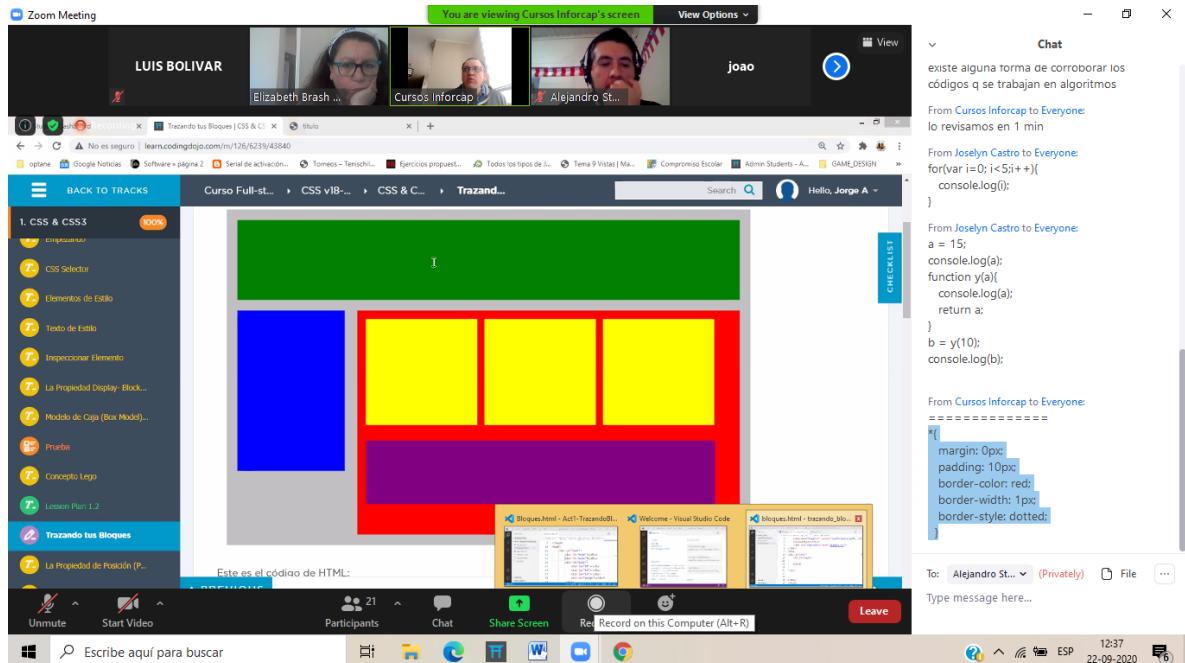
## AHORA EL DIV DE CABEZA



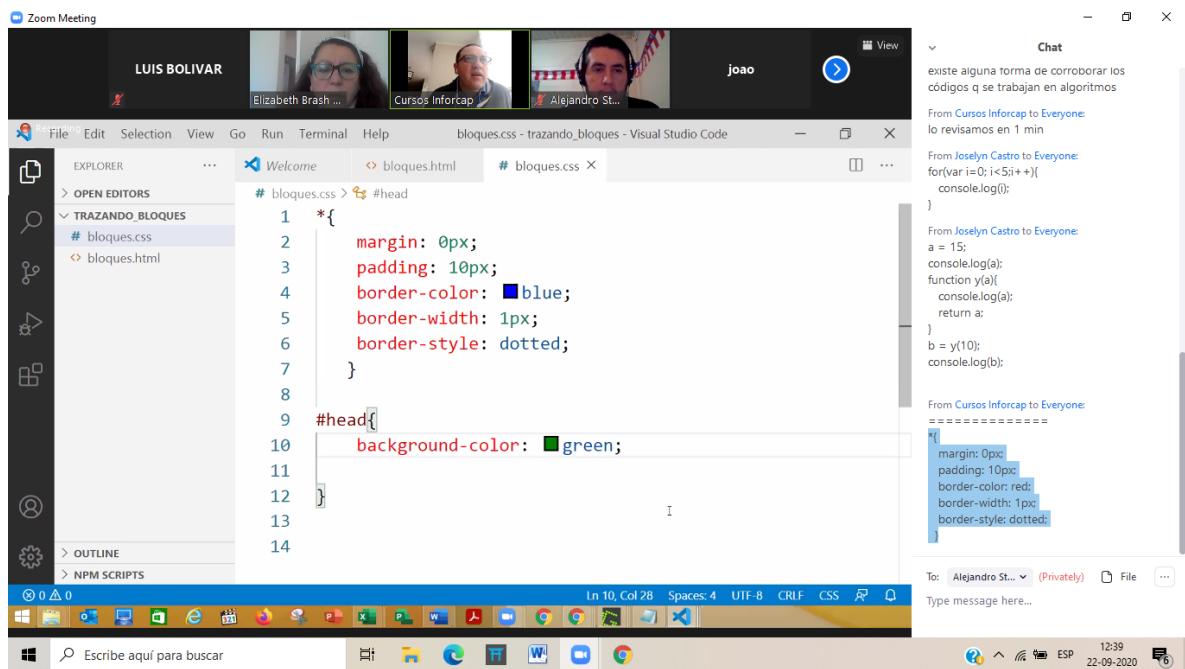
ASI SE VA VIENDO



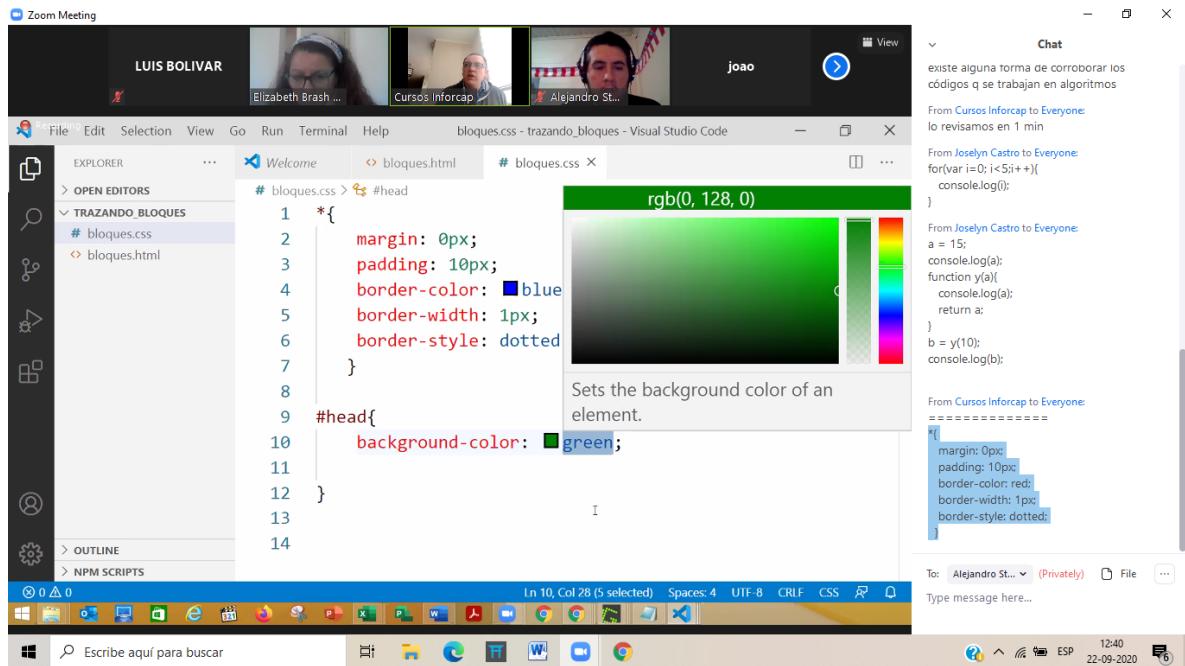
ENTONCES EL COLOR DEL HEAD VERDE



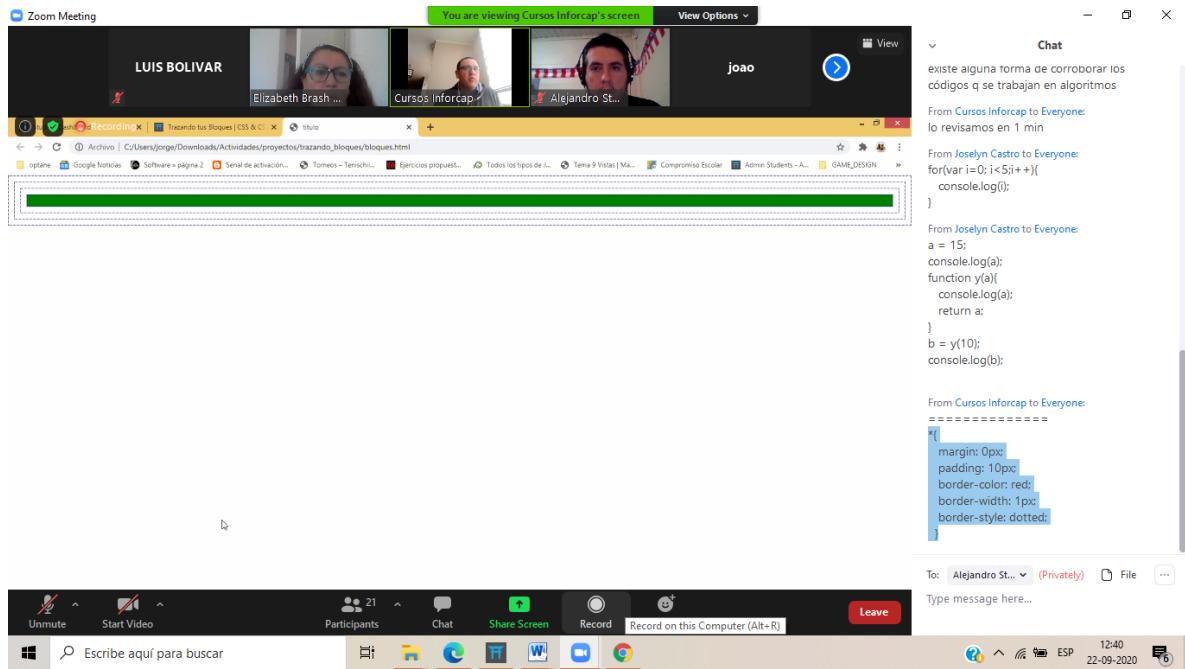
## #HEAD



PASANDO POR ENCIMA TAMBIEN SE PUEDE SELECCIONAR EL COLOR

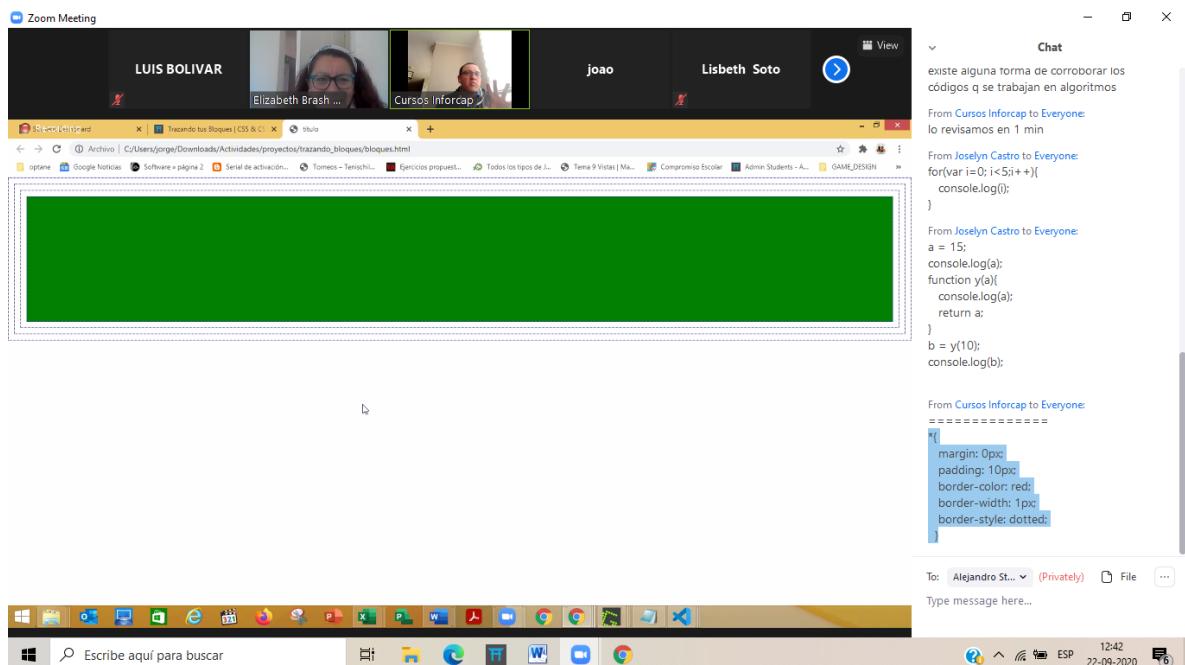
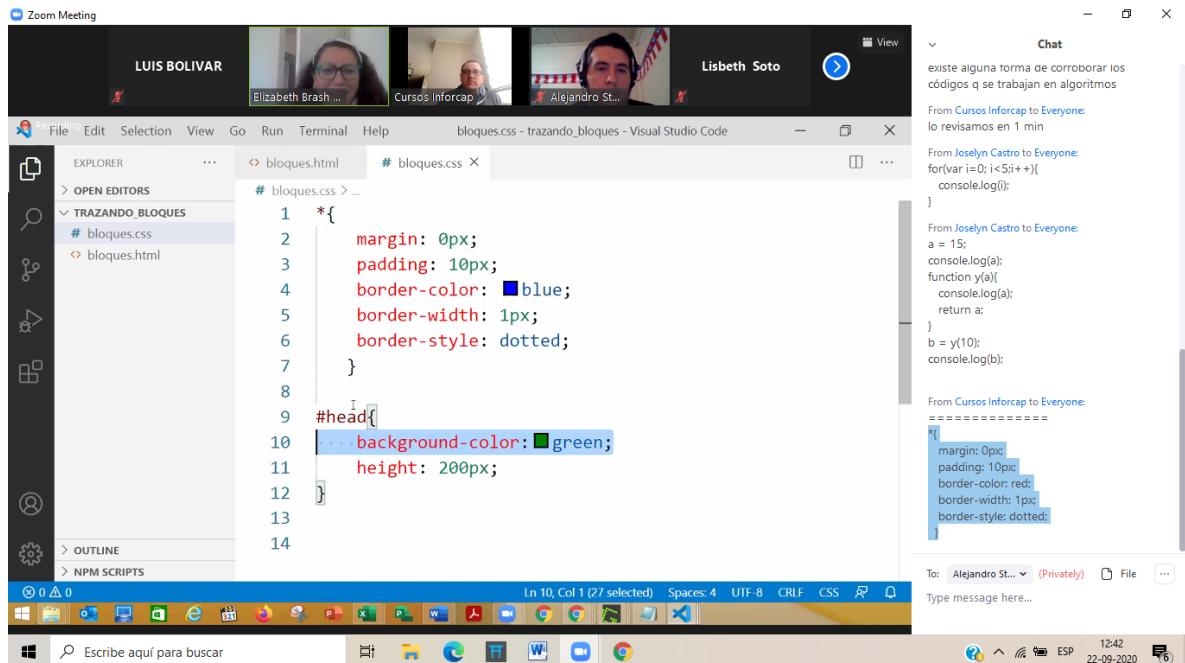


ENTONCES SE VEE

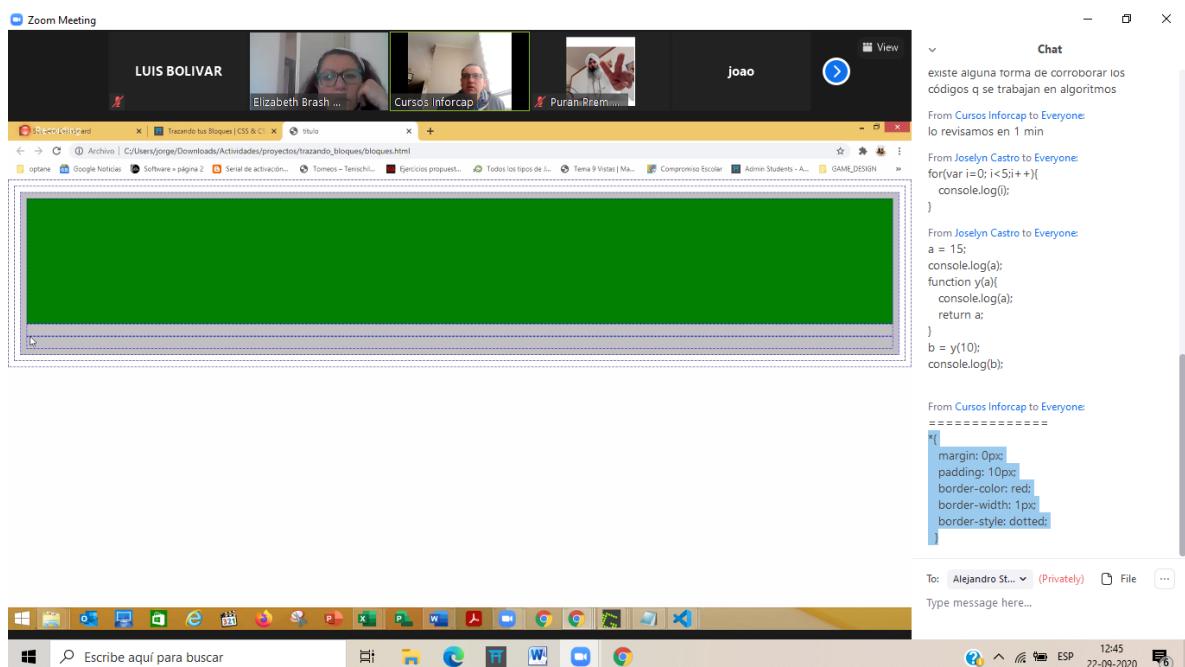
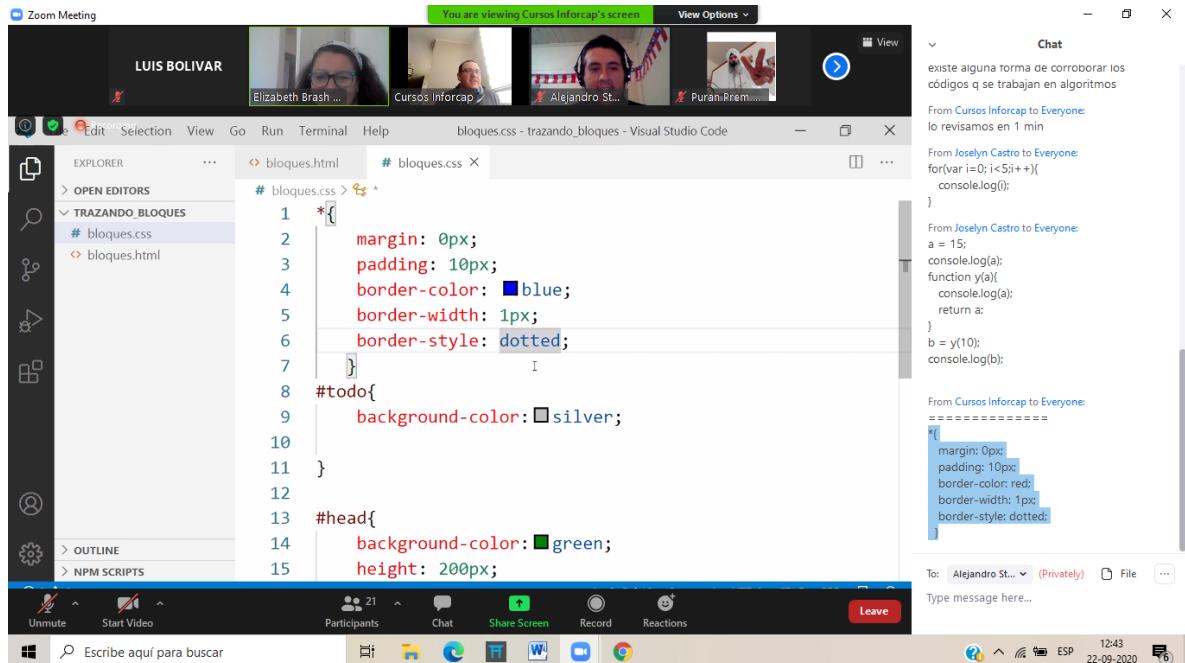


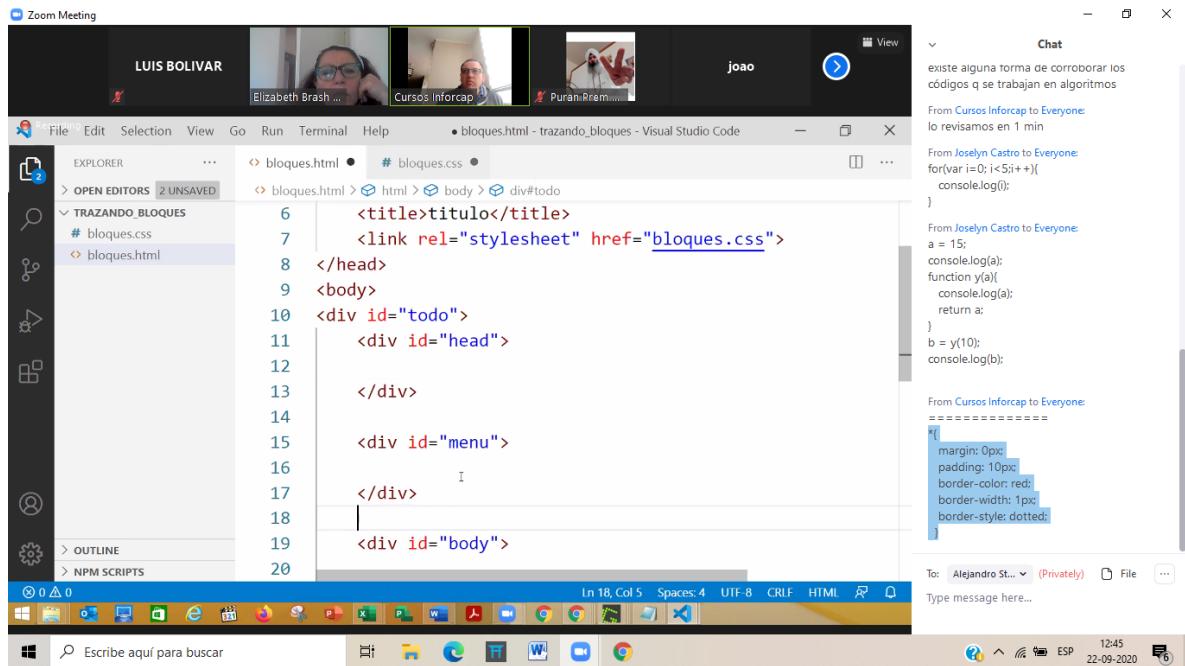
PERO HAY QUE VER EL ALTO

EN PIXELES

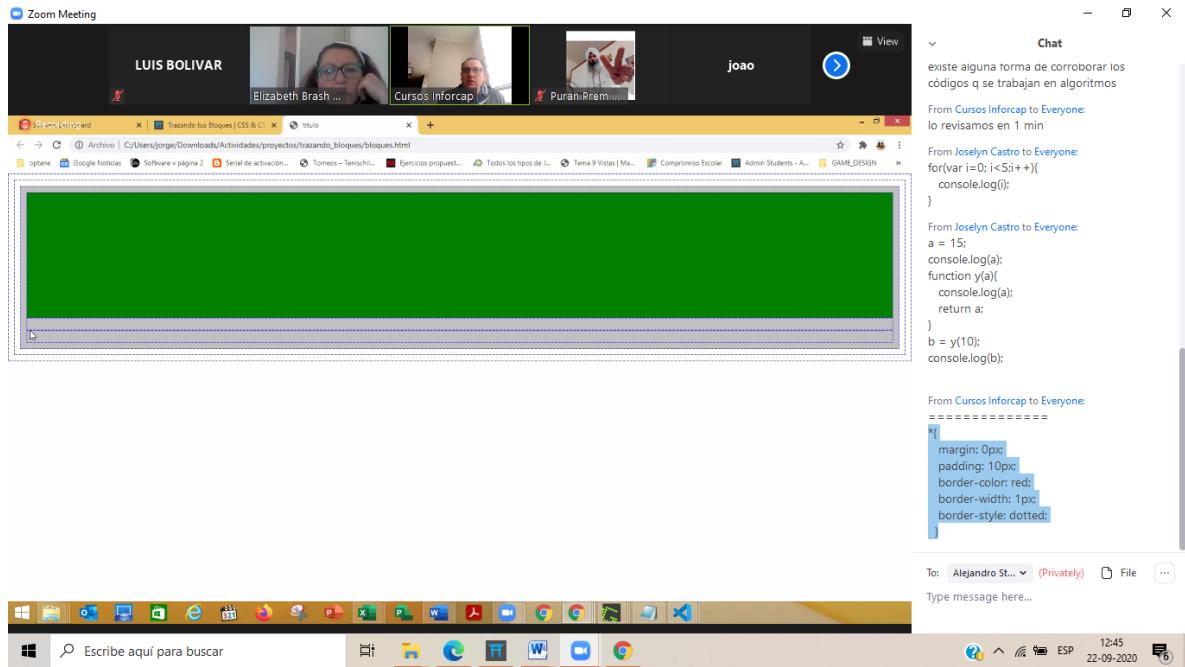


GRIS DEL FONDO





SE VEE



AHÍ ESTAN PERO HAY QUE ALINEAR Y DAR COLOR

**Zoom Meeting**

LUIS BOLIVAR    Elizabeth Brash ...    joao    Lisbeth Soto

File Edit Selection View Go Run Terminal Help bloques.css - trazando\_blocos - Visual Studio Code

EXPLORER    OPEN EDITORS

TRAZANDO\_BLOQUES

- # bloques.css
- bloques.html

```

13     background-color: green;
14     height: 200px;
15 }
16 #menu{
17     background-color: blue;
18 }
19
20 #body{
21     background-color: red;
22 }
23
24

```

Ln 21, Col 26    Spaces: 4    UTF-8    CRLF    CSS    🔍

Escribe aquí para buscar

Zoom Meeting

LUIS BOLIVAR    Elizabeth Brash ...    joao    Lisbeth Soto

File Edit Selection View Go Run Terminal Help bloques.css - trazando\_blocos - Visual Studio Code

EXPLORER    OPEN EDITORS

TRAZANDO\_BLOQUES

- # bloques.css
- bloques.html

```

9     background-color: silver;
10 }
11
12 #head{
13     background-color: green;
14     height: 200px;
15 }
16 #menu{
17     background-color: red;
18 }
19
20 #menu{
21     background-color: red;
22 }
23

```

Ln 16, Col 1 (selected)    Spaces: 4    UTF-8    CRLF    CSS    🔍

Escribe aquí para buscar

Chat

existe alguna forma de corroborar los códigos q se trabajan en algoritmos

From Cursos Inforcap to Everyone:  
lo revisamos en 1 min

From Joselyn Castro to Everyone:  
for(var i=0; i<5;i++){  
 console.log(i);  
}

From Joselyn Castro to Everyone:  
a = 15;  
console.log(a);  
function y(a){  
 console.log(a);  
 return a;  
}  
b = y(10);  
console.log(b);

From Cursos Inforcap to Everyone:  
=====

```

*{
margin: 0px;
padding: 10px;
border-color: red;
border-width: 1px;
border-style: dotted;
}

```

To: Alejandro St... (Privately)

Type message here...

12:49 22-09-2020

Chat

existe alguna forma de corroborar los códigos q se trabajan en algoritmos

From Cursos Inforcap to Everyone:  
lo revisamos en 1 min

From Joselyn Castro to Everyone:  
for(var i=0; i<5;i++){  
 console.log(i);  
}

From Joselyn Castro to Everyone:  
a = 15;  
console.log(a);  
function y(a){  
 console.log(a);  
 return a;  
}  
b = y(10);  
console.log(b);

From Cursos Inforcap to Everyone:  
=====

```

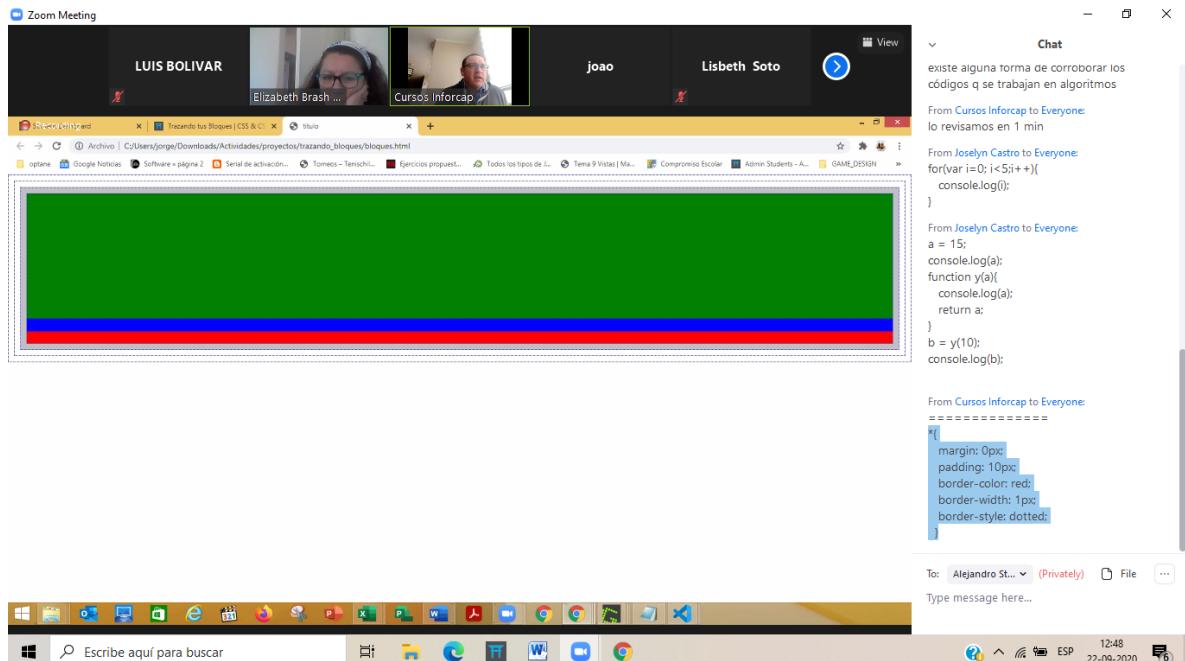
*{
margin: 0px;
padding: 10px;
border-color: red;
border-width: 1px;
border-style: dotted;
}

```

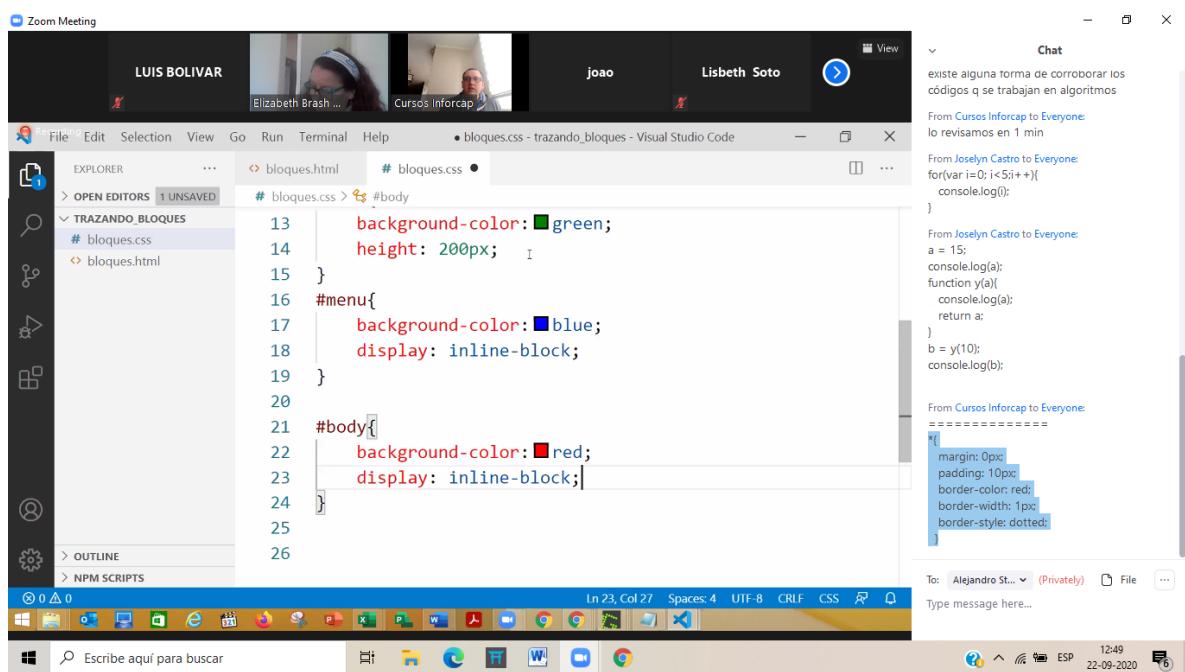
To: Alejandro St... (Privately)

Type message here...

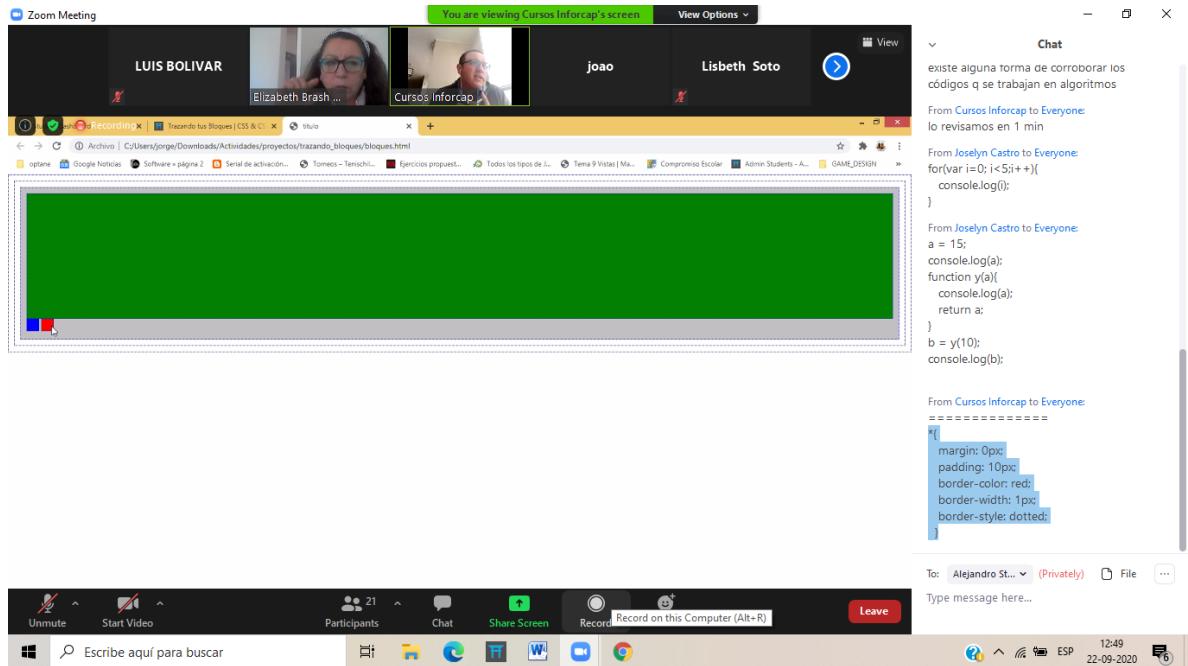
12:47 22-09-2020



## ALINEAR



## EN LINEA



AGRANDARLO

**Zoom Meeting**

File Edit Selection View Go Run Terminal Help bloques.css - trazando\_blocos - Visual Studio Code

EXPLORER OPEN EDITORS # bloques.html # bloques.css # body

```

17 background-color: blue;
18 display: inline-block;
19 width: 200px;
20 }
21 #body{
22 background-color: red;
23 display: inline-block;
24 width: 755px;
25 }
26
27
28

```

Ln 25, Col 15 Spaces: 4 UTF-8 CRLF CSS

To: Alejandro St... (Privately) File ... Type message here...

12:50 22-09-2020 Chat

existe alguna forma de corroborar los códigos q se trabajan en algoritmos

From Cursos Inforcap to Everyone:  
lo revisamos en 1 min

From Joselyn Castro to Everyone:  
for(var i=0; i<5;i++){  
 console.log(i);  
}

From Joselyn Castro to Everyone:  
a = 15;  
console.log(a);  
function y(a){  
 console.log(a);  
 return a;  
}  
b = y(10);  
console.log(b);

From Cursos Inforcap to Everyone:  
=====

```

*{
margin: 0px;
padding: 10px;
border-color: red;
border-width: 1px;
border-style: dotted;
}

```

To: Alejandro St... (Privately) File ... Type message here...

12:50 22-09-2020 Chat

You are viewing Cursos Inforcap's screen View Options

Zoom Meeting

LUIS BOLIVAR Elizabeth Brash ... joao Lisbeth Soto

Recording | Frecuencia tus bloques | CSS & C | 9:00

Participants 21 Chat Share Screen Reco Record on this Computer (Alt+R) Leave

Escribe aquí para buscar

12:51 22-09-2020 Chat

existe alguna forma de corroborar los códigos q se trabajan en algoritmos

From Cursos Inforcap to Everyone:  
lo revisamos en 1 min

From Joselyn Castro to Everyone:  
for(var i=0; i<5;i++){  
 console.log(i);  
}

From Joselyn Castro to Everyone:  
a = 15;  
console.log(a);  
function y(a){  
 console.log(a);  
 return a;  
}  
b = y(10);  
console.log(b);

From Cursos Inforcap to Everyone:  
=====

```

*{
margin: 0px;
padding: 10px;
border-color: red;
border-width: 1px;
border-style: dotted;
}

```

To: Alejandro St... (Privately) File ... Type message here...

12:51 22-09-2020 Chat

MAS RAPIDO EN TIEPMPO

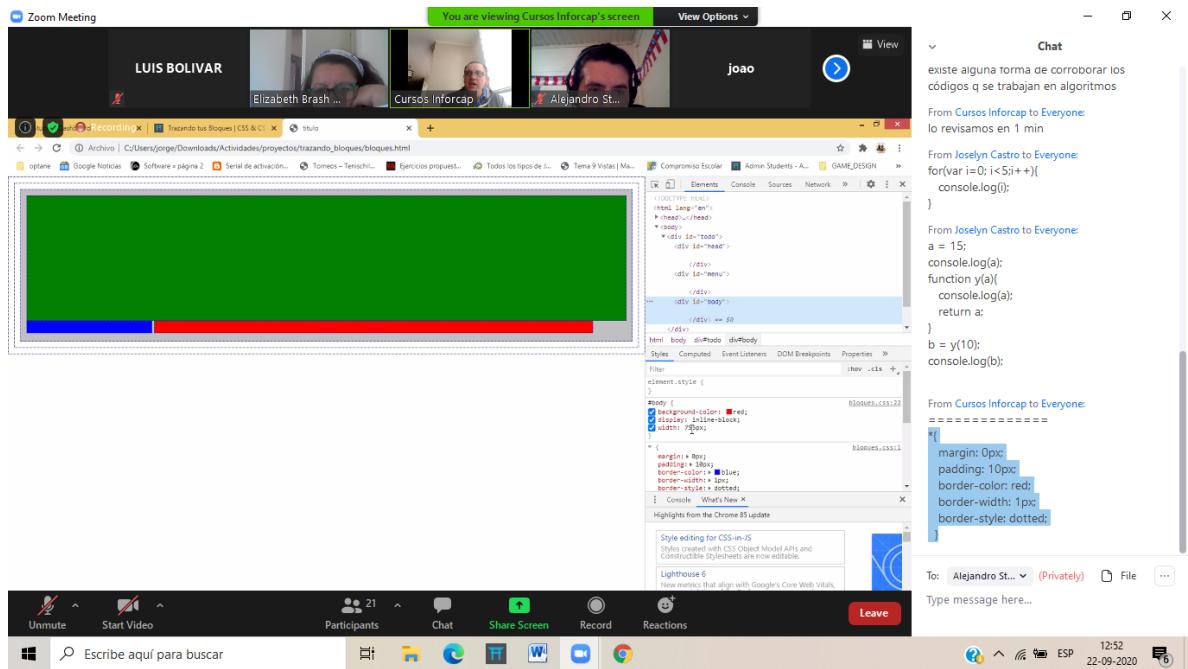
REAL

F12

ELEMENT

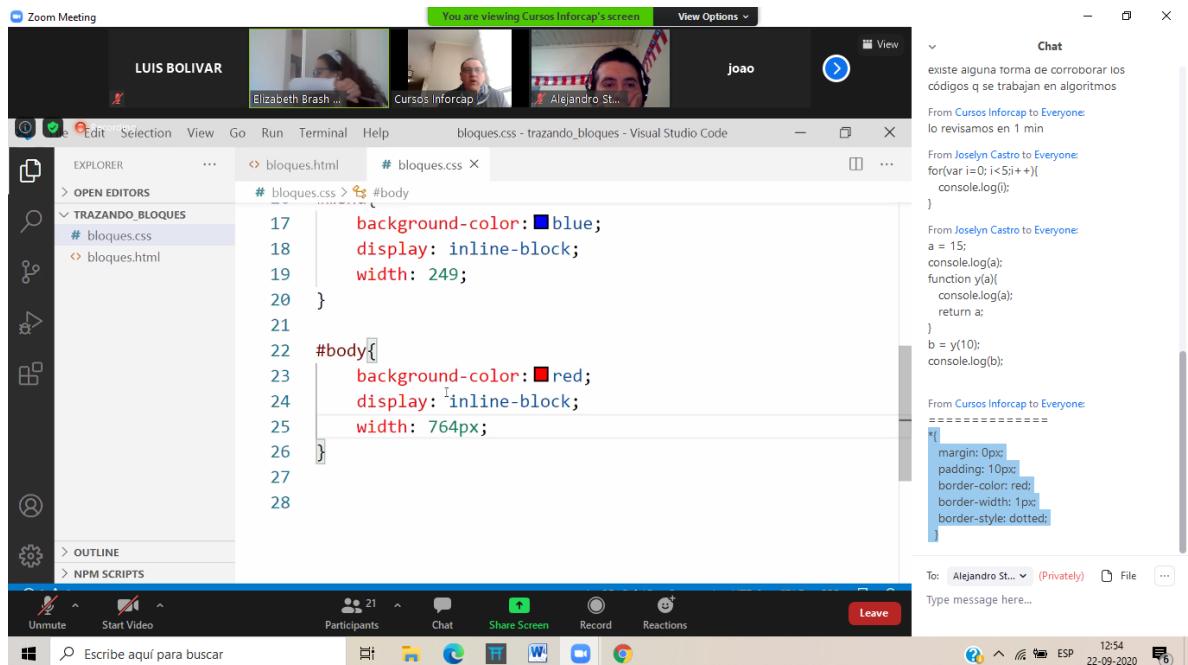
FLECHA AZUL LADO DERECHO

## MARCAR EL CUADRO



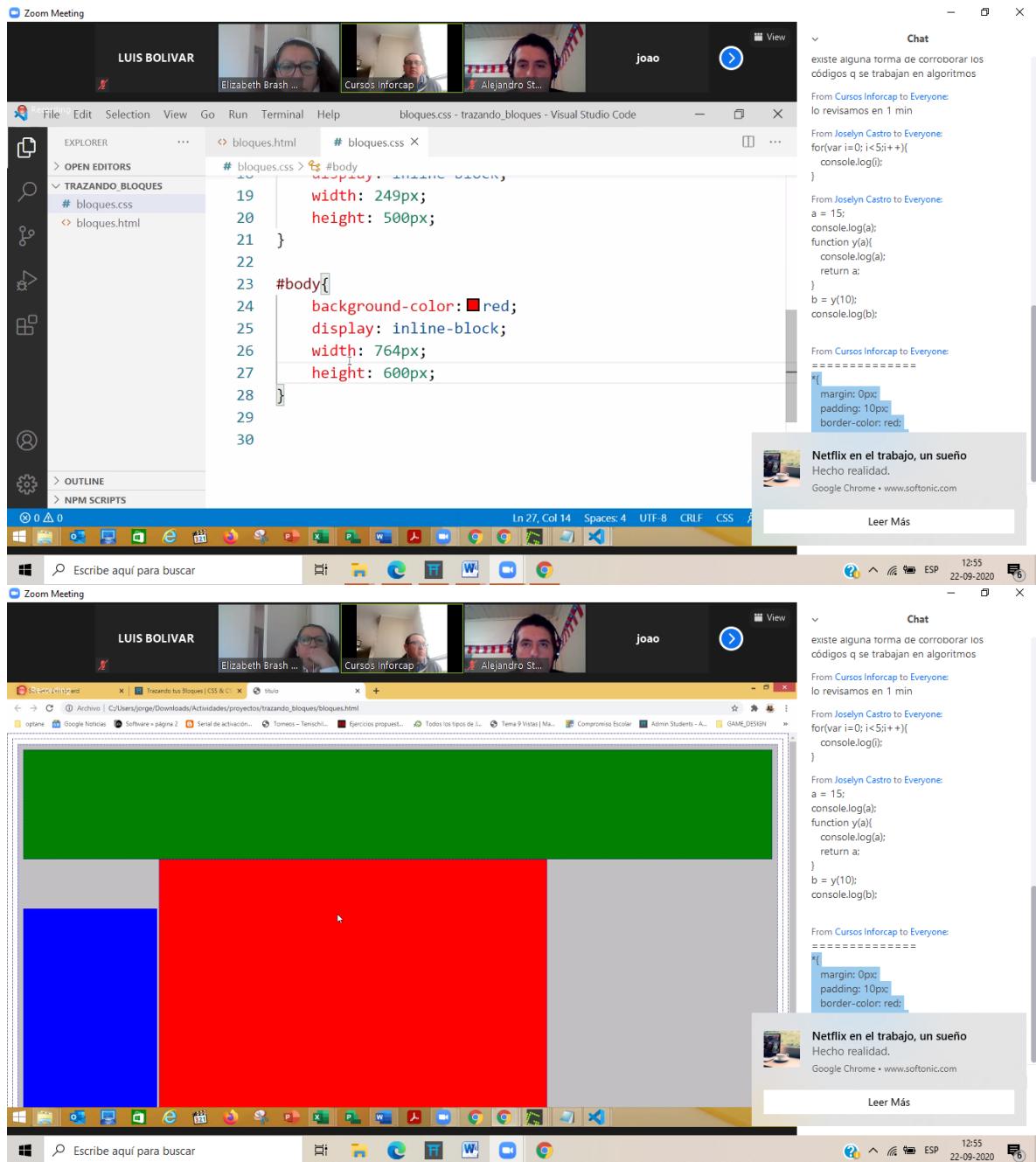
APRETA LAS FLECHAS Y VAN QUEDANDO DE INMEDIATO

LUEGO COLOCO EN LOS CODIGOS



COLOCAR 249 PX

PARA MAS LARGO



ESTA DECONFIGURADO

DAMOS ESPACIADO CUADRO AZUL PRIMERO VERTICAL

